

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 0 740 263 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
13.09.2000 Bulletin 2000/37

(51) Int Cl.7: **G06K 9/62**

(21) Application number: **96302899.8**

(22) Date of filing: **25.04.1996**

(54) Method of training character templates for use in a recognition system

Verfahren zum Trainieren einer Erkennungsanlage mit Zeichenmustern

Méthode pour entraîner un système de reconnaissance avec des modèles de caractères

(84) Designated Contracting States:
DE FR GB

(30) Priority: **28.04.1995 US 431223**

(43) Date of publication of application:
30.10.1996 Bulletin 1996/44

(73) Proprietor: **XEROX CORPORATION**
Rochester, New York 14644 (US)

(72) Inventors:
• **Kopec, Gary E.**
Belmont CA 94002 (US)
• **Chou, Philip Andrew**
Menlo Park CA 94025 (US)

• **Niles, Leslie T.**
Palo Alto CA 94306 (US)

(74) Representative: **Grünecker, Kinkeldey,**
Stockmair & Schwanhäusser Anwaltssozietät
Maximilianstrasse 58
80538 München (DE)

(56) References cited:
US-A- 5 321 773

• **IEEE TRANSACTIONS ON IMAGE PROCESSING,**
vol. 2, no. 4, 1 October 1993, pages 510-518,
XP000412065 KOPEC G E: "LEAST-SQUARES
FONT METRIC ESTIMATION FROM IMAGES"

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 740 263 B1

Description

[0001] The invention relates to a method and machine for training a set of character templates for use in a recognition system.

[0002] Character recognition systems typically include a process in which the appearance of an isolated input character image, or "glyph," is analyzed and, in a decision making process, classified as a distinct character in a predetermined set of characters. The term "glyph" refers to an image that represents a realized instance of a character. The classification analysis typically includes comparing characteristics of the isolated input glyph (e.g., its pixel content or other characteristics) to units of reference information about characters in the character set, each of which defines characteristics of the "ideal" visual representation of a character in its particular size, font and style, as it would appear in an image if there were no noise or distortion introduced by the image creation process. The unit of reference information for each character, typically called a "character template," "template" or "prototype," includes identification information, referred to as a "character label," that uniquely identifies the character as one of the characters in the character set. The character label may also include such information as the character's font, point size and style. A character label is output as the identification of the input glyph when the classification analysis determines that a sufficient match between the glyph and the reference information indicating the character label has been made.

[0003] The representation of the reference information that comprises a character template may be referred to as its model. One type of character template model is known as a bitmapped, or binary, image of a character. Within the category of binary character template models, at least two different types of models have been defined: one model may be called the "segmentation-based" model, and describes a character template as fitting entirely within a rectangular region, referred to as a "bounding box," and describes the combining of adjacent character templates as being "disjoint" - that is, requiring nonoverlapping bounding boxes. US-A-5,321,773 discloses another binary character template model that is based on the sidebearing model of letterform shape description and positioning used in the field of digital typography. The sidebearing model, described in more detail below in the discussion accompanying FIG. 1, describes the combining of templates to permit overlapping rectangular bounding boxes as long as the foreground (e.g., typically black) pixels of one template are not shared with, or common with, the foreground pixels of an adjacent template; this is described as requiring the templates to have substantially "disjoint supports."

[0004] Training character templates is the process of using training data to create, produce or update the templates used for the recognition process. Training data can be broadly defined as a collection of character image samples, each with an assigned character label identifying the character in the character set that it represents, that provide the information necessary to produce templates according to the character template model defining the templates. Training data, then, includes at least two components: (1) a glyph sample, and (2) its identifying character label. The effectiveness of existing training processes directly depends on the quality and accuracy of the training data, and specifically on the quality of the glyph samples, the correct identification of the character labels to be assigned to the glyph samples, and the assignment of correct labels to appropriate glyph samples.

[0005] Good quality glyph samples are those that are substantially unimpaired by missing or extraneous foreground pixels when they are input to the training process. Glyph samples derived from bitmapped images produced from well-known sources such as scanning and faxing processes are subject to being degraded by image noise and distortion which contribute to uncertainty in the actual appearance of the bitmap. A degraded bitmap appearance may be caused by an original document of poor quality, by scanning error, by image skewing, or by similar factors affecting the digitized representation of the image. Particular problems in this regard are the tendencies of characters in text to blur or merge, or to break apart. Such a degraded image will be referred to herein as a "noisy" image. The requirement of good quality glyph samples as an input to existing training processes has generally imposed the limitation that the input image used as the source of glyph samples be relatively non-noisy, or, if noisy images are permitted to be used, that there be some process for removing or otherwise compensating for the noise in the glyph samples.

[0006] Recognition systems typically provide distinct training subsystems for the purpose of training the character templates. Training systems may be "supervised" or "unsupervised." Unsupervised training generally involves a two step process of recognition and training.

[0007] In the context of this background discussion, existing supervised training is described as a process in which some aspect of the training data is, to some degree, specially prepared by a user for the training process. This may include either the isolation of the glyph samples, the identification of the character labels to be assigned to the glyph samples, or the actual assignment of the character labels to the glyph samples, or may include all three aspects. Supervised training provides for the opportunity to train new or existing templates using training data over which the user exercises some degree of control with respect to its quality and accuracy.

[0008] In one type of existing supervised training system, input glyph samples are required to be "segmented" - that is, isolated, individual, and relatively non-noisy glyph samples - and labeled with the proper character label prior to input to the training process. Typically, software with a user interface for training data preparation permits the user to manually draw bounding boxes around glyph samples in an image, and to assign labels to them, providing the user

with complete control of the quality and accuracy of the training data.

[0009] One or more of the three aspects of the preparation of the training data may be automated in order to reduce the amount of direct user involvement. For example, the segmentation of the glyph samples and determination of bounding boxes may be an automatic process applied to an entire text document image, or to an image of a line of text or to a word image. The user may have the opportunity to inspect the results of automatic segmentation, correct mis-segmented samples, and assign the character labels to the samples.

[0010] An image model is a characterization or description of the set of possible input images for which a recognition system is designed, provided in a form that can be used to determine which one of the possible images best matches a given input image. An image model represents a priori information about the set of possible input images and is distinguishable from data structures that define a particular input image or contain the results of performing analysis and recognition processing on a particular image.

[0011] A formal image model describes the set of possible images using a formal description language, such as a formal grammar or a finite state transition network. A formal grammar is a set of rules that define the allowable formats (syntax) that statements in a specific language are allowed to take. Grammars may be characterized by type as unrestricted, context-sensitive, context-free and regular, and a particular type of grammar may be more or less suited to a specific image model.

[0012] The design of every text recognition system is based on either an explicit or implicit image model. The distinction to be drawn is whether the image model is explicitly and formally stated in a manner that is independent of the processing algorithms that use the model, or whether the model is only represented implicitly, as a body of code that performs image analysis operations. A formal image model, in this regard, is analogous to a formal grammar in a grammar-based character string parsing system which exists as an explicit data structure independent of the code of the parser that uses it.

[0013] Formal image models may take zero-dimensional (0D), one-dimensional (1D) or two-dimensional (2D) forms.

[0014] H. S. Baird, in "A Self-Correcting 100-Font Classifier" (SPIE Vol. 2181 Document Recognition 1994) discloses an approach to the training of feature templates for a polyfont recognizer that uses a zero-dimensional image model. Baird discloses a self-correction method in which a polyfont classifier that is capable of recognizing any of 100 typefaces moderately well is able to specialize itself automatically to the single, but otherwise unknown, typeface that it is reading. The method requires a polyfont classifier capable of distinguishing among N character (symbol) classes, $\{q\}_{i=1,N}$, across a large number of typefaces with "reasonably good" accuracy, and further requires a classifier technology that is trainable on isolated sample character images labeled with classes. Baird's template training system is a form of unsupervised learning that requires that the image samples be isolated prior to inputting to the classifying and training processes.

[0015] S. Kuo and O.E. Agazzi, in "Keyword spotting in poorly printed documents using pseudo 2D hidden Markov models," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 8, August, 1994, pp. 842 - 848 (hereafter, Kuo et al.,) disclose an algorithm for robust machine recognition of keywords embedded in a poorly printed document. The templates, referred to as models, represent a set of known keywords that are to be matched. For each keyword model, two statistical models, named "pseudo 2D Hidden Markov Models", or "PHHMs," are created for representing the actual keyword and all other extraneous words, respectively. In the context of the terminology presented herein, the PHHM representing a keyword template is a formal 1D image model.

[0016] Feature-based template training in the context of a 1D image model for use in recognizing strings of characters (e.g., words), independent of character boundary information, is disclosed in C. Bose and S. Kuo, "Connected and degraded text recognition using hidden Markov model," in Proceedings of the International Conference on Pattern Recognition, Netherlands, Sept. 1992, pp. 116-119. The recognition method disclosed therein assumes that page-level processing using known algorithms is done prior to the training step and that isolated word or line images are presented to the recognizer, which is based on a formal 1D model expressed as a hidden Markov model. A supervised training process is disclosed that appears to be based on a 0D model of training individual feature-based templates each expressed as a hidden Markov model.

[0017] US-A-5,020,112 and 5,321,773 disclose recognition systems based on formal 2D image models. US-A-5,020,112, entitled "Image Recognition Using Two-Dimensional Stochastic Grammars" and issued to P. A. Chou, one of the inventors herein, discloses a method of identifying bitmapped image objects using a 2D image model based on a 2D stochastic, context-free grammar. US-A-5,020,112 discloses an object template library that contains a number of $n \times m$ bitmapped templates of all of the image objects of possible interest, each having an associated probability of occurrence in the image. The image glyphs are required to be substantially segmentable in the sense that their bounding boxes may not overlap significantly. The formal 2D image model is represented as a stochastic 2D grammar having production rules that define spatial relationships between objects in the image according to this rectangular image model; the grammar is used to parse the list of objects to determine the one of the possible parse trees that has the largest probability of occurrence. The training process is unsupervised, with the recognition process isolating the image samples to be used for training from an original input image, and assigning character labels to the segmented image

samples based on their classification by the recognizer. Because the template model is a segmentation-based model, in the context of the terminology used in this discussion, the 2D image model describes a set of possible images that each must contain substantially segmentable image objects, each of which must be able to be substantially contained in a nonoverlapping bounding box.

[0018] US-A-5,321,773 discloses a formal 2D image model represented as a stochastic finite state transition network that defines image generation in terms of a regular grammar, in contrast to the context-free grammar used in US-A-5,020,112. The template model described by the 2D image model defines the sidebearing model of letterform shape description and positioning, described in detail in the discussion accompanying FIG. 1.

[0019] Training of the character templates used in US-A-5,321,773 involves estimating or computing specific typographic characteristics, or parameters, that are required for proper template positioning; these are known as character sidebearings and baseline depths, collectively called font metrics. The shape of a glyph is defined in terms of a local coordinate system aligned so that the typographic origin of the glyph is at (0,0), shown by crosses 2, 5 and 6 in FIG. 1. The character "set width" of a glyph is defined in terms of glyph origin positions, and is the vector displacement $\Delta = (\Delta x, \Delta y)$ from the glyph origin position of a first glyph to the point at which the origin of a second adjacent glyph is normally placed when imaging consecutive characters. In most Indo-European alphabets, including Roman, $\Delta x > 0$ and $\Delta y = 0$. In FIG. 1, the character set width of the letter "e" is denoted by displacement Δx . In other writing systems, however, Δx can be negative (e.g., Semitic) or Δy can be nonzero (e.g., Oriental glyphs.) When $\Delta y = \alpha \Delta x$ for some α , the glyph origins in a line of text are colinear and define the baseline 4 of the text line. The bounding box 3 of a glyph is the smallest rectangle, oriented with the character coordinate axes, that just encloses the glyph. It can be seen from FIG. 1 that a typographic glyph image origin position is not necessarily coextensive with an x,y position of the bounding box; FIG 1 shows glyph image origin position 5 for the glyph "e" outside bounding box 3, and glyph image origin position 6 for the glyph "j" inside bounding box 8. The left sidebearing is the horizontal displacement λ from the origin of the glyph to the left edge of the bounding box. Similarly, the right sidebearing is the horizontal displacement ρ from the right edge of the bounding box to the origin of the next glyph. One or both of the sidebearings can be negative.

[0020] US-A-5,321,773 discloses the training of the character templates at col. 11 - 17, and the training process is further described in G. Kopec, "Least-Squares Font Metric Estimation from Images," in IEEE Transactions on Image Processing, October, 1993, pp. 510 - 519 (hereafter, "Kopec, 'Font Metric Estimation'"). The supervised training technique disclosed used a specially-prepared input image, shown in FIG. 14 of the patent, and in Fig. 3 of Kopec, "Font Metric Estimation" in which the glyph samples were segmentable. The samples were subjected to a pre-training segmentation step described at pg. 516 in Kopec, "Font Metric Estimation" in which the text lines and individual characters within each line of a font sample page were extracted using simple connected component-based analysis procedures of a text image editor. Each glyph sample isolated by the text image editor was labeled using a manually prepared text transcription of the sample page that included ordered character labels identifying the samples, paired on a one-for-one basis with the glyph samples in the input image.

[0021] The supervised training technique disclosed in US-A-5,321,773 and in Kopec, "Font Metric Estimation" is a manually intensive procedure. It requires that glyph samples be segmentable in the image source in which they occur, while the template model of the templates being trained requires that a pair of adjacently positioned character images of the templates need only have substantially disjoint supports. It would appear, then, that some document images having character images positioned according to the sidebearing model could not themselves be used as sources of glyph samples for training, and glyph samples for training would need to be specially prepared. Without glyph image origin position data, the font metrics may only be estimated using this procedure because information from which to measure them more accurately - the x,y coordinate glyph image origin positions of the sample glyphs - is not directly available from the training data.

[0022] It is the object of the invention to provide an improved method and machine for training a set of character templates.

[0023] This object is solved by the subject matters of independent claims 1, 4 and 8.

[0024] Preferred embodiments are defined in the dependent claims.

[0025] The present invention is based on the premise that a font-specific recognition system provides consistently higher recognition accuracy than a polyfont recognition system for documents that include character images that are predominantly in a specific font. To achieve maximum generality, polyfont recognition systems, also called "omni-font" systems, typically attempt to perform recognition using feature-based templates without particular sensitivity to specific typefaces, fonts or their type style variations. However, insensitivity to font can reduce accuracy if it prevents a recognition system from adapting to text that is predominantly or entirely in a single font, and the recognition accuracy of a single recognition system may vary widely for different documents each having a different font or for a single document having multiple fonts. Despite their generally higher degree of accuracy, however, font-specific recognition systems have been considered to be of limited usefulness and generality primarily because such systems were not easily able to be generalized to new fonts as a consequence of the need to train templates in each new font, and the need, in turn, for considerable user involvement in the preparation of the training data for the template training process; the

higher accuracy of such systems has been perceived to be outweighed by reduced productivity from their use when a new template set needed to be trained.

[0026] The present invention is based on the further premise that font-specific recognition systems could be a viable alternative to omnifont recognition systems if template training methods and systems existed that could accurately train templates for specific fonts while minimizing user intervention or involvement in the template training process without sacrificing user control over the quality and accuracy of the training data.

[0027] The present invention minimizes the user's involvement in all three aspects of training data preparation by using a two-dimensional (2D) image of a text document as the source of the glyph samples to be used for training; by using an unrestricted form of a transcription as a source of information about the labeling of the glyph samples; and by using a formal 2D image model as an explicit input to the training process that defines the relationship between the glyph samples in the 2D image and the information in the transcription so that substantially correct character labels are assigned to appropriate glyph samples. In fact, the training technique may be implemented in a manner that eliminates substantially all user involvement in training data preparation apart from providing the 2D image source of glyphs and a transcription associated with that 2D image, effectively resulting in the templates being automatically produced.

[0028] The use of a formal 2D image model as an explicit input to the training process provides the opportunity to use images of existing text documents as the source of glyph samples for template training, thus eliminating the need for a user to manually design and prepare a specific image of samples. The formal 2D image model represents a priori information about the set of possible 2D input images that are accepted as input to the training process. An important benefit of using an explicit formal 2D image model is that it provides for flexibility and detail in describing the set of possible 2D input images from which glyph samples for training are obtained, which in turn means that a wide variety of existing text document images may be used for training. For example, the model may include descriptions for non-glyph portions of the input image which do not contain glyph samples and therefore do not participate in the training process, thus permitting the use of text document input images having graphical or photographic images, or image portions of constant content such as that found in forms. A significant and unique advantage of the training technique of the present invention over existing training techniques is that the use of a text document image as the source of glyph samples for training eliminates the need for prior segmentation or isolation of glyph samples or lines of text in the 2D input image. The training process uses the information provided by the 2D image model to locate the positions of glyph samples in the input 2D image, thus reducing user involvement in training data preparation in supervised training systems where glyph sample segmentation is normally done manually by a user from a document image.

[0029] In the training technique of the present invention, the formal 2D image model, and specifically the character template model for the templates being trained, determines whether the text document image source of glyph samples must include segmentable glyph samples. When the formal 2D image model describes the template model as being a segmentation-based model, the 2D image model describes a set of possible 2D input images in which the glyph samples are segmentable, and for which bounding boxes for the glyph samples are required not to overlap. When the formal 2D image model describes the template model as being a non-segmentation-based model such as the side-bearing model described above, the 2D image model may describe a set of possible 2D input images in which the glyph samples need not be segmentable, and in which bounding boxes for the glyph samples are allowed to overlap, thereby further expanding the class of existing text document images that may be used as possible sources of the glyph samples.

[0030] A further benefit of using a 2D formal image model that is explicitly defined as an input to the training procedure is that the type (e.g., structural appearance) of text document images that may be used as the source of glyph samples for training may be changed by simply changing the formal 2D image model to reflect information about a new type of image; there is no need to rewrite the instructions that carry out the training process when the type of input image changes. The present invention places the detailed information about the input 2D image source of glyph samples in an input data structure that, in a particular implementation, may be made accessible to a user.

[0031] The use of a transcription in a flexibly defined and unrestricted form as an input to the training process provides the opportunity for the user to exercise explicit control over the labeling of the glyph samples that are used in template training, without requiring the user to explicitly prepare a specific transcription or to explicitly assign character labels to specific glyph samples. The training technique may be implemented in such a manner as to permit the user to prepare a literal transcription that results in correct character labels being assigned to specific glyph samples, but the technique may be implemented in a much more general manner so as to permit the user to merely select a suitable transcription that contains the information needed by the formal 2D image model to map character labels to glyph samples.

[0032] Another important advantage of the training technique of the present invention over existing training techniques is that the present invention provides for the use of a wider range of transcription types for training purposes than the one-for-one sequence of character labels used in conventional supervised training systems. In its simplest form, the transcription may be a string of transcription labels each indicating a character label, and each of which respectively pairs with a glyph sample in the 2D input image in a one-for-one pairing. The transcription may also contain

markup information, known as tags, that identify structural pieces of the document for a document processing, formatting or word processing application; this type of transcription is referred to herein as a "tag transcription."

[0033] Therefore, in accordance with the present invention, a method of operating a machine to train a plurality of character templates is provided. The machine operated by the method includes a memory device for storing data, including instruction data, and a processor connected for accessing the data stored in the memory, and for executing the instructions to operate the machine. The method uses as input an image definition data structure defining a two-dimensional image source including a plurality of glyphs, referred to as a 2D image source of glyph samples, having a vertical dimension size larger than a single horizontal line. Each glyph included in the 2D image source of glyph samples is an image instance of a respective one of a plurality of characters in a character set, referred to as a glyph sample character set. Each one of the plurality of character templates being trained represents a respective one of the plurality of characters in the glyph sample character set, and is identified by a character label data item indicating the respective character in the glyph sample character set. The method also uses as input a two-dimensional image source model data structure, referred to as a 2D image source model, that models as a grammar the spatial image structure of a set of two-dimensional (2D) images. The 2D image source of glyph samples is one of the set of 2D images modeled by the 2D image source model. The 2D image source model includes spatial positioning data that models the spatial positioning of the plurality of glyphs occurring in the 2D image source of glyph samples, and includes mapping data that maps a respective one of the glyphs occurring in the 2D image source of glyph samples to the glyph label indicating the character in the glyph sample character set. The template training method further uses as input a transcription data structure, referred to as a transcription, that is associated with the 2D image source of glyph samples and that includes an ordered arrangement of transcription label data items, hereafter referred to as transcription labels. The method comprises operating the processor to determine at least one glyph sample occurring in the 2D image source of glyph samples using the spatial positioning data included in the 2D image source model. The method further comprises operating the processor to produce a glyph label data item, referred to as a glyph label, to be paired with the at least one glyph sample occurring in the 2D image source of glyph samples, that indicates the respective one of the characters in the glyph sample character set represented by the paired glyph sample; the processor uses the mapping data included in the 2D image source model and the transcription to pair the glyph label with the at least one glyph sample. The method then includes operating the processor to produce one of the plurality of character templates indicating a respective one of the characters in the glyph sample character set using the at least one glyph sample occurring in the 2D image source of glyph samples identified by the paired glyph label that matches the character label of the bitmapped character template. Examples of the labeled glyph samples produced by the present invention include isolated, labeled character images; bounding boxes specified around glyph samples in image sources, each labeled with a character label; and 2D regions of the input image with origin positions of glyph samples identified, each labeled with a character label.

[0034] In accordance with another aspect of the present invention, a machine for use in training a plurality of character templates for use in a recognition operation is provided that comprises a first signal source for providing image definition data defining a first image; image input circuitry connected for receiving the image definition data defining the first image from the first signal source; a second signal source for providing non-image data; input circuitry connected for receiving the non-image data from the second signal source; a processor connected for receiving the image definition data defining the first image from the image input circuitry and for receiving the non-image data from the input circuitry; and memory for storing data. The data stored in the machine memory includes instruction data indicating instructions the processor can execute, and the processor is further connected for accessing the data stored in the memory. The processor, in executing the instructions stored in memory, receives from the image input circuitry an image definition data structure defining the 2D image source of glyph samples previously described and receives from the input circuitry the transcription data structure previously described. The processor, further in executing the instructions, receives from the input circuitry a two-dimensional image source model data structure, referred to as a 2D image source model, previously described. The processor, further in executing the instructions, performs operations that carry out the method steps previously described for determining the plurality of glyph samples occurring in the 2D image source of glyph samples, and for producing the glyph label data item paired with each respective one of the plurality of glyph samples that indicates a respective one of the characters in the glyph sample character set. The processor, further in executing the instructions, then produces the plurality of character templates using the glyph samples paired with respective ones of the glyph labels, as previously described.

[0035] An important advantage of the training technique of the present invention is its specific application to the training of character templates defined by a template model, such as the sidebearing model of letterform shape description and positioning, that requires identification of the origin positions of the character images in addition to or in place of identification of the bounding boxes of the character images. Training character templates based on the sidebearing character template model may be effectively accomplished knowing only the image origin positions of the glyph samples occurring in the 2D image, and need not also depend on determining a bounding box around a glyph sample to identify the pixels that are to be included in a particular character template. One implementation of the training

technique of the present invention, therefore, is a two step process; the first step uses the formal 2D image model and the transcription to identify the image origin positions of glyph samples in the 2D input image and to assign character labels to the glyph sample image origin positions. This implementation determines the image origin positions of the glyph samples without having any information in advance of training about where the glyph samples occur in the 2D input image.

[0036] In the second step of this implementation, the labeled glyph origin positions that are the output of the first step are then input to a novel template construction process that produces the trained templates. In existing training systems that train binary character templates, segmentation of the glyph samples through the use of bounding boxes allow for a relatively straightforward determination of the character template from multiple, isolated samples of the character, typically by using a well-known pixel averaging process. In the training technique of the present invention, only glyph origin positions of the glyph samples are determined, and segmentation of glyph samples in the 2D input image by determining bounding boxes around them need not be done. Therefore, existing bitmap averaging techniques that depend on knowing glyph sample boundaries cannot be used. The technique of the present invention produces a binary character template from information about only the image origin positions of the glyph samples using a novel template construction technique that essentially combines the functional result of glyph segmentation with the actual construction of the bitmapped templates. In this technique, an array of template pixel positions, called a template image region, having vertical and horizontal dimensions suitable for storing a binary character template, is produced for each character template to be trained, and used to determine, for each character template, sample image regions in the 2D input image, each containing one of the glyph samples for the respective character template. A template image region includes a template pixel position designated as the template origin position, and a sample image region in the 2D input image containing a glyph sample is determined relative to the local coordinate system of the template image region such that the image origin position of a glyph sample has the same relative pixel position in the sample image region as the pixel position of the template origin position in the template image region. All of the sample image regions indicating the same respective one of the characters in the glyph sample character set is called a collection of sample image regions, and each sample image region in the collection is aligned with all others, and with the template for that character, at the image origin position. The binary character templates are then produced substantially contemporaneously from the collections of the aligned sample image regions for each character template by assigning foreground pixel color values to template pixel positions in selected ones of the template image regions that are selected on the basis of template contribution measurements computed using respectively paired aligned sample pixel positions included in sample image regions.

[0037] There are several benefits and advantages that result from the training technique of the present invention as applied to training character templates described by a character template model that uses character image origin positions for character positioning. The explicit formal 2D image model that provides the a priori information about the 2D input image source of glyphs minimizes interference from noise in the input image in the process of locating the glyph image origin positions of the glyph samples. Moreover, elimination of a segmentation step prior to template construction eliminates segmentation errors introduced when performing segmentation on a noisy image, and so permits images of existing text documents of varying quality to be used as sources of glyph samples. The novel template construction process successfully handles blurred, broken and merged glyph samples that occur in noisy images, or merged glyph samples that naturally occur in a font in which character images have been designed to be connected. Since the template construction process uses multiple glyph samples that occur in the text of the 2D input image and produces the templates substantially contemporaneously, extraneous or missing dark pixels that occur in one or two blurred, broken or merged samples resulting from noise in the image are less likely to affect the quality of the character templates trained when these low quality samples are processed with other non-noisy and higher quality samples. Moreover, since this novel template construction method does not require that an actual bounding box for each glyph sample be found, a potentially wider range of symbol sets, alphabets and character fonts that do not easily lend themselves to segmentation may be accommodated by the training technique of the present invention, which in turn permits a wide variety of images of existing text documents to be used as sources of glyph samples.

[0038] In one implementation of the training technique of the present invention, the formal 2D image model, which describes the structure of a set of possible images that may serve as the source of glyph samples for training, is represented as a Markov source having the form of a finite state transition network that includes a series of nodes and transitions between pairs of nodes. Associated with each transition in the network are transition data items that include a character template, a message string, a transition probability and a vector displacement. A set of transcriptions, from which information about the character labels to be assigned to the glyph samples can be derived, is also represented as a finite state transition network in which each transition corresponds to a possible transcription label for the alternative transcriptions. One or more transcription labels are related to the message string(s) associated with transition(s) in the formal 2D image model such that the formal 2D image model together with the input set of transcriptions describes a relatively small set of possible input images. The training technique uses the formal 2D image model to find the set of transitions that present the best path through the 2D image model as constrained by the set of transcriptions. The

image origin position of each glyph sample along with a respective character label for that sample is identified from the sequence of transitions that form the best path, and this labeled glyph image origin positions data is then used as input to the novel template construction process described above.

[0039] The present invention will be described further, by way of examples, with reference to the accompanying drawings, in which:-

FIG. 1 illustrates a simplified version of the sidebearing model of letterform shape description and positioning;
 FIG. 2 illustrates a 2D image of a plurality of glyphs for use as a source of glyph samples for training character templates according to the present invention;
 FIG. 3 illustrates an example of a character template data structure produced by the present invention;
 FIG. 4 illustrates an example of a transcription data structure for the 2D image of FIG. 2 suitable for use as input to the training technique and system of the present invention;
 FIG. 5 illustrates another example of a transcription data structure for a portion of a dictionary that is suitable for use as an input transcription to the training technique of the present invention;
 FIG. 6 shows the 2D image of the dictionary page for which FIG. 6 is the transcription;
 FIG. 7 illustrates a set of transcription data structures representing alternative transcription messages for the 2D image of FIG. 2;
 FIG. 8 illustrates the set of transcriptions shown in FIG. 7 represented as a portion of a simplified, finite state transition network;
 FIG. 9 illustrates a formal 2D image source model represented in its general form as a simplified, finite state transition network;
 FIG. 10 is a simplified block diagram illustrating the input and output data structures of the training technique and system of the present invention;
 FIG. 11 is a flow chart illustrating the general steps of the character template training technique of the present invention;
 FIG. 12 is a simplified block diagram illustrating the input and output data structures of a portion of an illustrated software implementation of the training technique and system of the present invention that produces the training data input to template construction;
 FIG. 13 is a flow chart illustrating the general steps of the portion of the illustrated implementation of the present invention that produces labeled glyph image origin positions as training data;
 FIG. 14 is a diagram illustrating a finite state transition network modeling a class of 2D images having a spatial structure of a single text column, such as the 2D image of FIG. 2, used in the illustrated implementation of the character template training technique of the present invention;
 FIG. 15 is a diagram illustrating a simplified portion of the transcription network of FIG. 8 used in the illustrated implementation of the character template training technique of the present invention;
 FIGS. 16, 17, 18 and 19 schematically illustrate the merging of the finite state transition network of FIG. 14 with the transcription network of FIG. 15, according to the illustrated implementation of the present invention;
 FIG. 20 is a flow chart illustrating the decoding step in the flowchart in FIG. 13 as a Viterbi decoder, according to the illustrated implementation of the present invention;
 FIG. 21 illustrates the general steps of the template construction technique used in the illustrated implementation of the present invention;
 FIG. 22 illustrates the concept of a template image region used for storing a trained template during the template construction technique illustrated in FIG. 21;
 FIG. 23 illustrates the sample image regions that are identified in the 2D image source of glyph samples from which the templates are trained according to the template construction technique illustrated in FIG. 21;
 FIG. 24 is a schematic image of three sample image regions of the 2D image of FIG. 2, layered above the template image region of FIG. 22, illustrating the concept of sample image regions aligned at image origin positions of the glyph samples, according to the illustrated implementation of the present invention;
 FIG. 25 presents an image of a collection of sample image regions clipped from the 2D input image for use in training a template according to the illustrated implementation of the the present invention;
 FIG. 26 shows three exemplary but unsatisfactory templates produced using a technique that does not observe an important mathematical constraint imposed on character templates;
 FIG. 27 is a flow chart illustrating the steps for contemporaneously constructing all of the character templates using template image regions of FIG. 22 and aligned sample image regions of FIG. 24 and FIG. 25, according to the template construction technique of the illustrated implementation of the present invention;
 FIG. 28 shows a final set of trained templates produced according to the novel template construction technique illustrated in FIG. 27;
 FIG. 29 is a diagram illustrating a finite state transition network modeling a set of line images having a spatial

structure of a single text line and accommodating message strings consistent with an exemplary tag transcription; and

FIG. 30 illustrates sample line images that are modeled by the finite state transition network of FIG. 29.

[0040] The term "data" or "data item" refers herein to physical signals that indicate or include information. A first item of data "indicates" a second item of data when the second item of data can be obtained from the first item of data, when the second item of data can be accessible using the first item of data, when the second item of data can be obtained by decoding the first item of data, or when the first item of data can be an identifier of the second item of data. For example, directed arrow 36 in FIG. 3 shows that character label data item 28 in character template data structure 20 indicates character template 22, which depicts an image of the character "a." An item of data "identifies" or "is an identifier of" one of a set of identifiable items if the item of data is one of a set of items of data, each of which can be mapped to at most one of the identifiable items. For example, in FIG. 3, character label data item 28 may be said to identify character template 22.

[0041] Data defining an image may be referred to as "image definition data". For example, a two-dimensional (2D) array can define all or any part of an image, with each item of data in the array providing a value indicating the color of a respective location of the image. In this type of image representation, each such image location is conventionally called a "picture element," or "pixel," and represents a small unique region of the image. Typically, in black and white binary images, the value in a pixel indicates black or white, where black is the foreground color and is intended to represent a respective mark or active position in the image, and white is the background color. Because black is the typical color used as a foreground pixel color, reference to black pixels and foreground pixels will be used interchangeably throughout this discussion, unless explicitly noted otherwise. An image in a processor-controlled system that is represented by a 2D array of data items defining pixels is referred to as a "bitmapped image" or "binary image."

[0042] The term "display feature" refers to any human perception produced by a display device, and includes a single display feature and also may include plural display features that together form a pattern of display features in an image. A "display object" or "object" is a display feature that is perceptible as a coherent unity. An image "includes" a display feature or object if presentation of the image can produce perception of the feature or object.

[0043] "Character" as used herein means a single, discrete, abstract element or symbol. For example, a character can include an abstract symbol that appears in a written or printed form of a language. Characters in a language can include not only alphabetic and numerical elements, but also punctuation marks, diacritical marks, mathematical and logical symbols used in mathematical notation such as equations, and other elements used in the written or printed form of the language. More generally, characters can include phonetic, ideographic, or pictographic elements in addition to alphanumeric elements. For example, symbols in pictographic languages and symbols representing musical notation are included in the term character. All of the characters related to a particular language or other symbol notation such as music comprise a "character set."

[0044] A "character code" is an item of data in a processor-controlled machine or system that defines, or represents, a character (the abstract symbol) to the processor. The encoding of a set of characters, such as those that belong to a language, involves defining a set of character codes that includes a respective character code for each character in the set. An example of a set of character codes is the set of ASCII codes for the symbols that make up the English language.

[0045] A "glyph" is a single instance, or example, of a character that is realized as an image, for example on a marking medium such as paper or on a display screen. Because a variety of factors may affect how an image of a character is produced when it is printed, scanned, copied or faxed, one glyph of a character in a text image may not be identical to another glyph of the same character in the text image.

[0046] The terminology "image definition data defining an input 2D image source of a plurality of glyphs" (hereafter also referred to as a "2D image source of glyph samples," "2D image data structure," or simply as the "2D image") refers to a data structure, suitable for storage in a memory device of a processor-controlled machine, that defines a 2D image in which a plurality of bitmapped representations of characters occur in the 2D space defined by the image. The organization of the 2D image data structure is such that individual pixel locations are accessible by the processor, but the pixels that comprise an individual glyph are not initially identified as a unit of data that is accessible to the processor, and no information is initially available to the processor as to whether a specific x,y coordinate position in the 2D image indicates one of the pixels included in a glyph. The 2D image source of glyph samples is the input source of glyph samples used for training character templates according to the present invention. A 2D image is conceptually analogous to a page of a document, and may frequently represent an image of an actual physical page, with glyphs being vertically, as well as horizontally, distributed in the 2D space. The 2D image is not limited to include only glyphs; other image objects such as graphical objects or shapes, pictures, halftone images, line drawings, photographs, other pictorial elements, or images that constitute noise may be included in the input 2D image source of glyphs.

[0047] FIG. 2 illustrates 2D image data structure 10 that includes bitmapped representations of characters in the character set that comprises the English language. In FIG. 2, each discrete representation of an English language

character in 2D image 10 is a glyph; glyphs 12 and 14 have been enlarged to illustrate a schematic representation of the individual pixels that make up their respective images. 2D image 10 in FIG. 2 illustrates a portion of the data structure representing a binary image that has been produced by scanning a newspaper article, and includes pixels comprising line segment 16, a nonglyph, included in the 2D image.

[0048] A "template", or "character template", is a data structure that indicates a bitmapped image of a character. The "support" of a bitmapped character template is the set of pixel locations where the template differs from the background. A "character label" is a data item that indicates information uniquely identifying one of the characters in a character set with the respective character template indicating the bitmapped image of the character. A character label may indicate a character code, such as an ASCII code, to identify the template, or may indicate some other information that uniquely identifies the template as the one indicating the bitmapped image of a particular one of the characters in a character set, such as font identifying information, size, or type style information. A "set of labeled character templates" or a "set of character templates" is a data structure that includes at least one character template and the respective character label that uniquely identifies the character template.

[0049] FIG. 3 illustrates set 20 of labeled character templates representing characters in the English language character set. Character template data structures 22, 24 and 26 each indicate character label data items 28, 30 and 32, respectively, as shown via exemplary directed arrow 34 from character template 22 to character label 28. Identifying information in each of character label data items 28, 30 and 32 is shown as a character in quotation marks; this representation is used in the figures herein to indicate a respective character code stored in a data memory of a processor-controlled machine, as distinguished from pixels that represent an image of the character.

[0050] The illustration of character templates in FIG. 3 is not intended to limit the data structure that represents a character template in any way to an explicit 2D array of pixels representing a complete character. A template may be constructed from the concatenation of pieces of bitmapped characters, such as vertical strokes, joins, ascenders, descenders, and curved portions. A template may also be represented by a formal model that produces an explicit 2D array of pixels representing a complete character as its output.

[0051] A "transcription data structure" or "transcription" as used herein is a data structure indicating a unique message string, M. Message string M includes a plurality of message substrings, $m_1, m_2, \dots, m_n$, each of which indicates at least one of a plurality of characters in a character set. Each substring, m_i , is referred to as a "transcription label data item," or simply as a "transcription label." A transcription is said to be "associated with" a formal 2D image source model (defined below) when the formal 2D image source model together with the information indicated by the characters in the transcription establishes a mapping between one or more glyph samples in an input 2D image source of glyphs and one or more character labels indicating character templates in the set of character templates being trained. The term "mapping" is used herein in its mathematical sense to refer to a rule of correspondence established between two sets that associates each member of the first set with a single member of the second. Transcription labels have no implied or inferred order with respect to one another, or with respect to the glyph samples in the associated 2D image unless the transcription is of a type in which the order of the transcription labels is explicitly indicated by the definition of the transcription type.

[0052] A transcription is said to be "associated with" a specific input 2D image source of glyphs when the transcription data structure meets one of two conditions: (1) The transcription data structure is, or is capable of being produced from, the output of a recognition operation performed on the input 2D image. The recognition operation may be processor-controlled, such as a computer-implemented recognition or decoding operation performed on the specific 2D image. Or the recognition operation may be performed by a user using a processor-controlled machine; for example, a user may produce the transcription data structure by visually inspecting the 2D image and entering character codes using a conventional input device, such as a keyboard, that produces signals indicating the character codes. (2) The transcription data structure is, or is capable of being produced from, a data structure that is an input source to an image rendering operation, such as a document formatting operation, that produces the 2D image. The input 2D image with which a transcription is associated is referred to as the "associated 2D image."

[0053] A "literal transcription" is a type of transcription that includes an ordered sequence of transcription labels, each of which indicates a character label for a character template in the set of templates being trained, and substantially all of which, taken in the sequential order of their occurrence in the transcription, can be paired, by visual inspection of an associated input 2D image, with respective individual glyphs occurring in the associated image that represent the characters indicated by the respective character labels in the transcription, when the glyphs are taken in a sequence consistent with the reading order for the associated 2D image. FIG. 4 illustrates a literal transcription 60 that is associated with 2D image source of glyphs 10 (FIG. 2) and that includes a single ordered sequence of transcription labels. Newline character 62 is a label that indicates that the character labels following the newline character have paired glyph samples that are positioned in the next line of the associated 2D image; newline characters are commonly inserted in data structures indicating text by users preparing a text document using a text editor.

[0054] A "nonliteral transcription" is a type of transcription associated with an input 2D image source of glyphs that is not a literal transcription. A nonliteral transcription may include, for example, a transcription label that is not an error

and that indicates a character that cannot be paired by visual inspection with a glyph in the associated 2D image; or the non-character-label data may indicate characters in a character set other than the character set represented by the templates being trained. For example, a special character, mathematical symbol, or a musical notation that appears as a glyph in an associated input 2D image may be represented in a transcription label as a character in an extended character set or as a string of one or more ASCII characters. A nonliteral transcription may intentionally omit transcription labels for some glyphs in an associated 2D image. An example of this type of transcription is one associated with a 2D image of a preprinted form, where the transcription contains the transcription labels for the information that is inserted into the fields of the form, but omits transcription labels for what appears as the preprinted information on the form, such as the graphical elements and the glyphs that provide instruction information.

[0055] A "tag transcription" is a type of nonliteral transcription of an associated 2D image source of glyphs in which the non-character-label data indicates information, called a "tag," or "tag data," that, when interpreted by a document processing operation, produces display features that are perceptible in the format of, or as part of the glyphs in, the associated 2D image. Tag data includes information that identifies format characteristics of the 2D image such as page, paragraph and line breaks and line spacing that is deterministic of the location of a glyph in the 2D image; that specifies one or more appearance attributes of one or more glyphs, such as the font or font variation in which a glyph appears; or that results, when the document is formatted, in producing glyphs in the 2D image to which no explicit transcription label in the transcription can be mapped. The various types of information that may be indicated by a tag will be referred to generally herein as "markup information." Tag data typically occurs in data structures intended to serve as standardized document interchange formats for representing document structure and content between document processing systems; such data structures are typically produced as the output of an operation that converts either an input document data structure or an input image to a data structure representing the document in the standard interchange language. Examples of such languages include SGML (Standard Generalized Markup Language), ODA (Office Document Architecture) and HTML (Hypertext Markup Language). Tag data also occurs in data structures used or produced by document specification and formatting systems, also called automatic text layout systems, that include within the data structure instructions for the format and logical structure of a document, such as found in document data structures produced using markup languages. Examples of such document specification and formatting systems include GML (Generalized Markup Language), TeX and LaTeX.

[0056] FIG 5 illustrates tag transcription data structure 40 for a dictionary page of a technical dictionary; tag transcription 40 is associated with the image of the dictionary page shown as 2D image 50 in FIG. 6. Tag transcription 40 includes transcription labels 42, 44, 46 and 48 indicating tag data that identify the structural parts of an entry, name, category and body, respectively, of a dictionary entry.

[0057] FIGS. 5 and 6 also illustrate an example of tag data that results in the generation of glyphs in the 2D image to which no explicit transcription label in the transcription can be mapped. Image 50 includes left and right brackets 54 and 55 respectively, around a sequence of glyphs depicting the characters "electr". It can be seen that tag transcription 40 does not include transcription labels for left and right brackets 54 and 55, but that it does include transcription label 47 indicating that message substring "electr" is category information for the word entry. It can be inferred that message substring "\category", when interpreted as formatting information by a document processing operation, produced right and left brackets 54 and 55 as display objects in image 50.

[0058] A "set of transcriptions" refers to at least two transcriptions of the same type, all of which are associated with a single 2D image. Mathematically, the set of transcription data structures is considered to be a regular set. FIG. 7 illustrates the straightforward case of a single transcription message providing alternative transcription label 72 of "F (r l n) (om l orn)" for the glyphs representing "From" in input 2D image source of glyphs 10 (FIG. 2), producing a set of four transcription data structures 70 for image 10.

[0059] A "formal transcription model," or "transcription model," is a data structure that represents the message string of the transcription as a regular set, such as a finite state transition network or a grammar. For example, a finite state transition network has a number of transitions, each transition corresponding to a transcription label in the transcription.

[0060] A set of transcriptions may be represented as a formal transcription model. Model 800 in FIG. 8 illustrates an example of a finite state transition network for the set of transcriptions 70 in FIG. 7, illustrating the transitions that occur for the word "From" in transcriptions 70.

[0061] A "formal two-dimensional image source model," or "formal 2D image model," is a data structure that defines a mapping between the glyph positions of the glyph samples in the 2D image and character labels of templates that identify the glyphs as samples of the characters indicated by respective character labels. The formal 2D image model is an explicit input to the training technique and system of the present invention, and contains instructions, in the form of a formal description language such as a formal grammar or a finite state transition network, that characterize or describe a priori information, including structural features and functional characteristics, about the set of possible 2D input images for which a recognition system is designed and a set of possible transcriptions that may be associated with the set of possible images. The formal 2D image model also describes a character template model that models the templates to be trained, and includes an initial set of character templates. The formal 2D image model is analogous

to a formal grammar in a grammar-based character string parsing system which exists as an explicit data structure independent of the instructions (i.e., the code) of the parser that uses it.

[0062] The formal 2D image model enables the present invention to operate in the 2D image domain so as to require no pre-training step of text line isolation or individual glyph segmentation. Structurally, a formal 2D image model intended to be the type of model suitable for use in the training technique and system of the present invention defines image position information about how possible image objects (e.g., glyphs, graphical objects, photographs, etc.) in an image included in the set of possible 2D images are spatially arranged in the 2D image. Frequently, but not necessarily, when the image represents an English language document, glyph position information defined by the model is consistent with the conventional reading order for the document when the document is read by a human. In the illustrated embodiment described below, images, including input 2D image source of glyphs 10 in FIG. 2, are assumed to be rectangular, and to have an image coordinate system 13 (FIG. 2) in which x increases to the right, y increases downward, and the upper left corner is at $x = y = 0$. The model's description of image position information for non-glyph image objects permits a portion or portions of a given input image to be eliminated as possible image positions of glyph samples. This aspect of the model permits a wide variety of input 2D images to be accommodated as glyph sample sources, and the model may be constructed to describe any one of a number of classes of input 2D images, including, for example, images of printed music, images of equations, and images with fixed or known structural features such as business letters, forms and telephone yellow pages.

[0063] The formal 2D image model expresses transcription label information as the set of rules that define the mapping between the information indicated by the message substrings in the transcription and actual message substrings, in terms of character labels, that appear in a 2D image, and, for each substring, what its appearance is in the image. This mapping effectively establishes a mapping between the set of possible 2D images, the set of possible transcriptions, and the set of character templates that enables the training technique to determine which one of the possible 2D input images - that is, which sequence of characters in which sequence of lines of text strings - best matches a specific 2D input image associated with a specific transcription. From this best match information, the model enables the training technique to determine the positions of the glyph samples in the 2D image, and to assign character labels to the samples. The specific position information about the glyph samples that needs to be determined by the training technique is a function of the particular template model that defines the character templates. The template model defines how character templates, and therefore glyph samples, are spatially arranged or positioned with respect to one another in an image. If the template is defined as a segmentation-based model, the training technique must be able to produce information indicating glyph bounding boxes from the mapping established by the formal model. If the template is defined as a non-segmentation-based model, such as the sidebearing model, the training technique must be able to produce information indicating glyph origin positions from the mapping established by the formal model.

[0064] The design of the formal 2D image model to be used as the input to the training procedure is influenced by the type and content of the transcription to be used, and so permits further flexibility to the user in providing training data to the training procedure. The information included in the formal 2D image model about the structural and functional features of the transcription is only that information needed by the model to establish the requisite mapping between glyph samples and character labels, which in turn is the additional information needed by the model to specify a specific image from the set of possible images defined by the model. The farther removed the information in the transcription is from a literal transcription of an associated input 2D image source of glyphs, the more information is needed in the 2D image model to establish the correct mapping.

[0065] An example of an implementation of the formal 2D image source model of the type intended to be used in the present invention, and that is used in the illustrated embodiment described below, is a stochastic finite state transition network that represents its production rules as a regular grammar, and that explicitly defines the sidebearing model of letterform shape description and positioning as its character template model. A simplified, general illustration of this model as a Markov source is schematically shown as model 820 in FIG. 9, and is described in more detail below in the discussion of a particular implementation of the present invention.

[0066] The character template training technique 200 of the present invention illustrated in the block diagram of FIG. 10 is provided with inputs of the 2D image source of glyph samples 10, a formal transcription model 810, and an image model 40, all of which have been defined above. Character template training technique 200 uses these input sources of data to produce character template data structure 20 which includes a set of character templates and their respective character label data items for a particular character set.

[0067] The present invention recognizes that the 2D image source of glyph samples typically contains multiple sample images of a unique character in a character set in a particular font, and that, if information indicating the 2D image x, y coordinate positions and the character identity of each sample were known, a trained character template for each unique character in the 2D image, in the specific font of the samples, could be derived from the pixel colors of the set of pixels that make up each glyph sample. The present invention also recognizes that a transcription associated with the 2D image provides identity and sequence information for each of the glyphs in the 2D input image that may be used to identify the character of a respective one of the glyph samples. The grammar-based 2D image source model,

explicitly specified as an input to the invention, defines spatial positioning information about the glyphs that occur in the 2D image source of glyph samples in order to locate the glyph samples, and defines mapping data indicating a mapping of a respective one of the glyphs occurring in the 2D image to a glyph label indicating the character in the glyph sample character set.

[0068] Character template training technique 200 is capable of producing a labeled character template only for those characters for which glyph samples occur in 2D image 10, and the completeness of the character set for which character templates are trained is dependent on the 2D image source of glyph samples 10 having at least one glyph sample for each character in the character set. For example, because the English letters "j," "q," "x" and "z" do not appear in the portion of 2D image 10 shown in FIG. 2, character template training technique 200 will not produce these templates when only that portion of 2D image 10 is used as the source of glyph samples. For convenience hereafter, the character set for which character templates are being trained will be referred to as the "glyph sample character set," to indicate its dependence on the glyph samples that occur in 2D image source of glyph samples 10. The quality of each trained template is generally dependent on the number of glyph samples available in the 2D image source of glyph samples 10 for each character in the glyph sample character set.

[0069] The general steps of the character template training technique 200, illustrated in FIG. 11, include, in box 220, determining the position of each glyph sample in 2D image 10 using the spatial positioning information defined by grammar-based 2D image source model 40, and determining, in box 250, a glyph label for each glyph sample located in 2D image 10 using transcription 70 and the mapping data defined by 2D image source model 40 mapping a respective one of the glyph samples occurring in 2D image 10 to the glyph label indicating the character in the glyph sample character set represented by the glyph sample. The result of steps 220 and 250 is to produce a data structure indicating a set of labeled glyph samples which is the training data from which character templates may be produced. Then, in box 270, character templates are constructed using the data structure indicating the set of labeled glyph samples. In a particular implementation of character template training technique 200, glyph samples and their labels may be determined contemporaneously, and the functions in steps 220 and 250 could be viewed as being combined to produce the training data that is input to step 270; this combined functionality is indicated in FIG. 11 by procedure 210 enclosing steps 220 and 250 in a dotted line box.

[0070] The organization and content of the output data structure indicating the training data produced by steps 220 and 250 may take one of several forms determined by several interrelated factors that reflect aspects of the particular implementation of character template training technique 200.

[0071] When the character template model of the character templates being trained is the sidebearing model, procedure 210 must produce training data that indicate labeled glyph samples, each of which is identified by an x,y position in 2D image source of glyph samples 10 that indicates the image origin position of the glyph sample in 2D image 10. A novel template construction technique, described in detail beginning with the discussion accompanying FIG. 21, is used in step 270 to construct the binary character templates using a list of labeled glyph image origin positions indicating image origin positions of glyph samples in 2D image 10.

[0072] When the character template model of the character templates being trained is a segmentation-based model, step 220 finds bounding boxes for each glyph sample in 2D image 10, and produces as training data a list of labeled bounding box coordinates, or a set of labeled, isolated glyph samples extracted from 2D image 10. Any well-known pixel averaging and thresholding technique may then be used in step 270 to produce the binary character templates from the segmented or isolated training data. In one such technique, the number of foreground and background pixels for each pixel location in each of the samples is counted and divided by the total number of samples, and a threshold value is used to evaluate whether the average should yield a foreground or background determination.

[0073] With reference to FIG. 12, 2D image source model 830, represented as a stochastic finite state transition network similar to that disclosed in U.S. Patent 5,321,773, and finite state transcription network 850 are inputs into network merge process 300 which produces a merged finite state network called a transcription-image network 870. This merged network is then used to decode 2D image source of glyph samples 10 using a Viterbi decoding process 330 that produces a best sequence of transitions, or path, through the merged network. An initial set of character templates 500 is used during this decoding process; dotted line 832 indicates that initial set of character templates 500 is part of finite state transition network 830 since, as noted earlier, character templates may be attributes on transitions in network 830. Process 374 identifies the transitions and their corresponding image origin positions in 2D image 10 that compose the best path through transcription-image network 870, as produced by Viterbi decoding process 330. Process 380 then determines the image origin positions and message strings from the transitions that have non-null character template attributes; these image origin positions indicate estimated locations of glyph samples in 2D image 10. The training data, i.e., labeled glyph image origin positions 390, are the output of this sequence of processes.

[0074] The flowchart in FIG. 13 illustrates the steps of an implementation of the character template training of FIG. 11 that uses finite state transition network 830 as the formal 2D image source model and transcription network 850 to represent the transcription. The decoding step 330 uses a current set of templates to determine the best path through transcription-image network 870. For the initial iteration of decoding, an initial set of character templates having arbitrary

content may be generated by the processor, for association with the transitions in transcription-image network 870. The decoding, training data extraction, template construction and character set width determination steps, in boxes 330, 380, 400 and 490, are iterated until a stopping condition is met, which is tested in box 384, and the set of templates used in decoding step 330 during iterations subsequent to the initial iteration is the current set of templates produced as the output of template construction step 400. Training data extraction from the best path produced by decoding 2D image 10, illustrated in FIG. 12 as processes 374 and 380, is shown as combined process 380 in FIG. 13. Processes 300, 330 and 380 are now discussed in more detail.

[0075] With reference to FIG. 9, the structure of a set of images is captured formally by modeling image generation as image source model 820, which is also called a Markov source. A Markov source consists of a finite set of states (nodes, vertices), N , and a set of directed transitions (branches, edges) B . Each transition t connects a pair of states, L_t , and R_t , that are called, respectively, the predecessor (left) state and the successor (right) state of t . Two distinguished members of N are the initial state n_i , denoted by reference numeral 822, and the final state n_f denoted by reference numeral 824. It is assumed that no transition has n_f as its predecessor, so that the final state is a trap state. With each transition t is associated a 4-tuple of attributes 826, $(Q_t, m_t, a_t, \Delta_t)$, where Q_t is the template, m_t is the message string, a_t is the transition probability, and Δ_t , denoted by reference numeral 828, is the vector displacement of t , analogous to set width for characters. (See the earlier discussion accompanying FIG. 1 for a description of the character set width.) In the illustrated implementation, some of these attributes may be null for particular transitions, each transition message string m_t of image source model 820 is assumed to be either the empty string ε or else contains a single character, and vector displacement 828 may have negative, zero or positive scalar component values. The template Q_t is defined over the entire image plane Ω , although normally its support (set of non-zero, foreground pixels) will be localized within a small region near the origin of the local template coordinate system.

[0076] A "path" in a Markov source is a sequence of transitions $t_1 \dots t_p$ for which $L_{t_1} = n_i$ and

$$R_{t_i} = L_{t_{i+1}} \quad (1)$$

for $i = 1, \dots, P - 1$. A "complete path" is a path for which $R_{t_p} = n_f$. A "cycle" or "loop" is a sequence of transitions $t_1 \dots t_p$ for which $L_{t_1} = R_{t_p}$. Associated with each path Π is a composite message, or message string,

$$M_\pi = m_{t_1} \dots m_{t_p} \quad (2)$$

formed by concatenating the message strings of the transitions of the path. A message string in this context is another term for the transcription, and the terms message, message string, transcription and transcription string are used interchangeably in the discussion of the illustrated implementation. The set of transcriptions generated by complete paths through a Markov image source is a regular language and the source model itself is a finite state automaton that accepts that language. A Markov image source model defines a probability distribution on complete paths by

$$\Pr \{n\} = \prod_{i=1}^P a_{t_i} \quad (3)$$

and induces a probability distribution on messages by

$$\Pr \{M\} = \sum_{\pi | M_\pi = M} \Pr \{n\} \quad (4)$$

with M_π being the message associated with a path Π .

[0077] Also associated with each path π is a sequence of vector image pixel positions $x_1 \dots x_{p+1}$

$$x_1 = 0 \quad (5)$$

$$x_{i+1} = x_i + \Delta_{t_i}, \quad (6)$$

where x_{p+1} is introduced for convenience, and a composite image Q defined by

$$Q_\pi = \bigcup_{i=1}^P Q_{t_i} \{x_i\} \quad (7)$$

where $Q[x]$ denotes Q shifted so that the origin of its local coordinate system is located at x , and the union of two template images is an image that has foreground pixels where either of the two template images has a foreground pixel. For a path Π ,

$$\Delta_\pi = x_{p+1} - x_1 = \sum_{i=1}^P \Delta_{t_i} \quad (8)$$

is defined to be the displacement of the path and Δx_π and Δy_π denote the x and y components of Δ_π , respectively. A pair (x_i, t_i) consisting of one of the positions defined by (5) or (6) and the corresponding transition of the Markov source will be called a "labeled transition image origin position." The set of all such pairs defined by a complete path is called the set of labeled transition image origin positions of the path. For each transition t , let N_t denote the number of transition image origin positions of the path that are labeled with t and let the corresponding transition image origin positions be denoted $x_1^{(t)} \dots x_{N_t}^{(t)}$. Thus,

$$P = \sum_{t \in B} N_t \quad (9)$$

[0078] Based on the premise that fonts are typically designed so that the foreground pixels of the character glyphs do not overlap (i.e., share the same foreground pixels) in text strings, image source models of the type illustrated in FIG. 9 and in this illustrated implementation are required to be designed so that the union of the pixels of a template positioned at (x_i, t_i) with the pixels of a template positioned at (x_j, t_j) is the empty set

$$Q_{t_i}[x_i] \cap Q_{t_j}[x_j] = \emptyset \quad (10)$$

for $i \neq j$, for every path Π . This requirement expressed in (10) may be referred to as the "template disjointness constraint" of neighboring template supports.

[0079] Image source model 820 (FIG. 9) defines a relation, or mapping, between message strings and images via an underlying path and (2) and (7) that is bi-directional.

[0080] An image source model defines a finite state acceptor for the language of messages generated by the model. Thus, given a message string M , it is straightforward to determine if there is a complete path Π for which $M_\Pi = M$, and, if such a path exists, to find one. The image Q_Π defined by (7) is then an image of M . If the image source model defines a deterministic acceptor for the message language, the process of message imaging using the image source model admits a simple procedural interpretation. Imagine an imager automaton (referred to as an "imager") that draws what may be called an "ideal" image in an output image plane under the control of an input message "program". The structure of the imager is defined by a finite state image source model of the type illustrated in FIG. 9. The imager begins at location $(0,0)$ of the output image plane in internal state n_1 . The imager examines the first character in the input message, compares it with the message labels on the transitions from n_1 , and selects the branch whose message matches the input character. If the template associated with the selected branch is non-null, the imager draws a copy of the template on the output image plane, with the template origin aligned with the imager's current image position. The imager then increments its image position by the branch displacement and updates its internal state to the successor node of the selected branch. This process is repeated for each character of the input message until ideal image Q_Π - and a path

through the network from initial node, n_i to the final node, n_F - is completed.

[0081] As an image decoder, image source model 820 may be used to extract simple text strings from an observed image to produce a literal text transcription of the image (i.e., a transcription without formatting or logical structure tags.) These text strings are extracted from the message string attribute associated with each transition included in a path identified through model 820 as the observed image is being decoded. Image source model 830 in FIG. 14 models a set of 2D images having the common spatial structure of a simple text column and will be used to illustrate the process of image decoding in more detail. A simple text column consists of a vertical sequence of text lines, alternating with white (background) space. Horizontally, a text line is a sequence of characters typeset according to the sidebearing model shown in FIG. 1. 2D image source of glyph samples 10 is a representative image of the type modeled by image source model 830. Model 830 models a path through a 2D image of a single column of text that follows the conventional reading order for a text in the English language, assuming that the path through the image starts at the top left corner of the image and proceeds to the bottom right corner, and proceeds from the left of the image to the right in repeated 1D line sequences. Each transition t_i between nodes in the network has the associated 4-tuple of attributes, shown in FIG. 14 in the order $[a_i]$ (Δ_i), m_i , Q_i , where, when a template Q_i is associated with a transition, the message string m_i identifies the character represented by the template. It can be seen that some of these attributes are null for some transitions.

[0082] With reference now to FIG. 14, state n_1 corresponds to the creation of vertical white space. Each time branch t_1 is traversed, the imager moves down one row without drawing anything on the output image plane, since no image template is associated with t_1 . At some point, the imager reaches the top of a text line and follows branch t_2 . The displacement (0,B) of t_2 moves the cursor down to the text baseline; B is the font height above baseline. State n_2 represents the creation of a horizontal text line. The self-transitions from n_2 to n_2 are of two types. The F transitions t_i that are labeled with image templates Q_i and single-character message strings " c_i " are used to draw individual glyphs on the output image plane. The horizontal displacement associated with each of these branches is the character set width, W_{ci} . Branches t_3 and t_4 have blank templates associated with them and represent white space. Branch t_3 represents a white space of minimal (1 pixel) width and is used for fine spacing adjustment. Branch t_4 corresponds to a real space character of font-dependent width W_s and is labeled with a space message " ". At the end of a text line, the imager traverses t_5 ("line feed") and enters "carriage return" state n_3 . The message on t_5 is the new line character "\n". The vertical displacement associated with t_5 is the font depth D. Each traversal of branch t_6 moves the imager left by one pixel. Finally, transition t_7 returns the imager to state n_1 and the process is repeated for the next text line. After the last text line has been created, the imager traverses t_8 into final state n_F .

[0083] Transcription data structure 70, which represents a set of possible transcriptions associated with 2D image source of glyph samples 10, is also represented as a finite state network, referred to hereafter as a "transcription network." Transcription network 850 is a simplified form of a finite state image source model of the type illustrated in FIG. 9 in which each transition is associated with a message string m_i , but no other attributes. FIG. 15 shows a simple example of a portion 852 of transcription network 850 for transcription 70, representing the set containing the two transcription strings "orn\n" and "om\n" where the symbol "\n" represents the new line character. In the illustrated implementation, as is the case with image source model 830, each transition message string m_i of transcription network 850 is assumed to be either the empty string ϵ or else contains a single character. The data structure representing transcription network 850 is a received and stored, in box 292, as an input to the template training method in the illustrated embodiment. Transcription network 850 may be produced from transcription data structure 70 by a prior manual or automated process, such as by a process that uses conventional tools for producing finite state string grammars and transition networks.

[0084] Image source model 830 and transcription network 850 jointly define an ideal image that is a spatial arrangement of copies of character templates placed at specified image locations in the ideal image and selected according to a message string consistent with the transcription, and this ideal image is an approximation of the actual input 2D image with which the transcription is associated. It follows from this that decoding 2D image 10 using image source model 830 would be most efficient if decoding were able to be constrained to generate only ideal images, and consequently paths, that were consistent with the paths, and consequently message strings, generated by transcription network 850. Such a constraint can be imposed on the decoding process that uses image source model 830 by merging image source model 830 with transcription network 850.

[0085] The inputs to the network merge step 300 (FIGS. 12 and 13) are 2D image source model 830 and transcription network 850. The output of this step is a second Markov image source model of the type illustrated in FIG. 9, called the transcription-image network 870. Transcription-image network 870 is defined by the following two properties: (a) for each complete path Π in the transcription-image network there is a complete path in image source model 830 which has the same transcription string and image as Π ; and (b) for each complete path Π in image source model 830, if the transcription of Π is in the set of transcriptions generated by transcription network 850, then there is a complete path in transcription-image network 870 which has the same transcription string and image as Π . The set of transcriptions generated by the transcription image network is the intersection of the set of transcriptions generated by image source

model 830 and the set of transcriptions generated by transcription network 850. The ideal images generated by the transcription-image network that have a given transcription are the same as those generated by image source model 830 having that transcription.

[0086] Network merging step 300 is essentially concerned with constructing transitions between pairs of image source and transcription network states in the merged transcription-image network such that the transcription-image network satisfies the two properties (a) and (b) defined above. These transitions are constructed according to the following three steps.

1. For each transition t of image source model 830 for which $m_t = \epsilon$ (i.e. the message associated with t is the null string) and for each $j=0 \dots T-1$: add to the transcription-image network a transition from node (L_t, s_j) to node (R_t, s_j) . The message, template and displacement associated with each such transition of the transcription-image network are the same as those of t .

2. For each transition t of image source model 830 for which $m_t \neq \epsilon$ (i.e. the message associated with t is a single-character string) and for each transition t' of transcription network 850 for which $m_{t'} = m_t$: add to the transcription-image network a transition from node $(L_t, L_{t'})$ to node $(R_t, R_{t'})$. The message, template and displacement associated with each such transition of the transcription-image network are the same as those of t .

3. For each transition t' of transcription network 850 for which $m_{t'} = \epsilon$ and for each $I=0 \dots N-1$: add to the transcription-image network a transition from node $(n_I, L_{t'})$ to node $(n_I, R_{t'})$. The message and template associated with each such transition of the transcription-image network are both empty, and the displacement is the vector 0.

[0087] The construction of a portion of transcription-image network 870 is schematically illustrated in FIGS. 16, 17, 18 and 19 using image source model 830 of FIG. 14 for the simple text column and portion 852 of transcription network 850 shown in FIG. 15. FIG. 16 illustrates the nodes of the transcription-image network constructed by network merge process 300 as dots or points in a two-dimensional (2D) lattice 860, with image source model nodes 862 positioned horizontally and transcription network nodes 864 positioned vertically in 2D lattice 860. The lattice points 866 and 868 for the initial state (n_I, s_I) and the final state (n_F, s_F) , respectively, are each represented by a circle around a dot. FIG. 17 shows the transcription-image network after constructing transitions in the transcription-image network according to step (1) of the above procedure. For simplicity, transition probabilities are not shown. Fig. 18 shows the transitions of FIG. 17 added in step (1) of the network merge process as dotted lines, and shows the transitions that are added to the transcription-image network in step (2) of the above procedure in solid lines. Again, transition probabilities and displacements are not shown. Because transcription network 850 in FIG. 15 contains no transitions with empty message strings, step (3) of the above procedure for constructing transitions is not applied in this example.

[0088] Any node that cannot lie on a complete path may be deleted from the combined transcription-image network prior to its use in decoding, as can all transitions that enter or leave a deleted node. FIG. 19 depicts the portion 872 of the combined transcription-image network that remains after this simplification is performed. Note that this simplified, or merged, network contains significantly fewer states and transitions than the combined transcription-image network of FIG. 18. Thus, network simplification, or merging, typically leads to faster decoding of the input source of glyph samples.

[0089] Decoding process 330 (FIG. 13) may be accomplished using any type of software- or hardware-implemented decoder suitable for decoding 2D image 10 using the merged transcription-image network to produce the labeled glyph image origin positions indicating the glyph samples in 2D image 10. In particular, a decoder based on a dynamic programming algorithm that minimizes the probability of error between the original input 2D image and a target ideal 2D image, Q_{II} , is likely to be the most appropriate decoding process to use for a particular implementation.

[0090] In general, a decoding process of the type suitable for use in the present invention will identify some or all of the complete transcription-image paths through the transcription-image network, each of which indicates a target ideal 2D image, Q_{II} , and will determine which one of the identified paths is the best path, by determining which target ideal 2D image best matches the 2D image source of glyph samples according to a defined matching criterion. The best path through the network is the transcription-image path that indicates the best-matched target ideal 2D image; transition image origin positions in the 2D image source of glyph samples can be computed from the transitions that make up this best path, and glyph image origin positions and their labels are available, in turn, from selected ones of the transitions and their transition image origin positions. The matching criterion may be any suitable image measurement; typically, the matching criterion will involve optimizing a pixel match score for the target ideal image compared to the 2D image source of glyph samples.

[0091] Decoding process 330 (FIG. 13) in the illustrated implementation finds the maximum a posteriori (MAP) path through the transcription-image network, using the assumed asymmetric bit flip channel model. The purpose of the Viterbi decoder is to maximize a recursive MAP decision function over all complete paths through the transcription-image network in order to determine the most likely path through the network. As noted above in the discussion of decoding using image source model 830, each complete path through the transcription-image network corresponds

to an ideal image produced during decoding. Thus, the Viterbi decoder determines which ideal image, of the possible ideal images produced from complete paths through the network, is closest in appearance (by pixels) to the input image being decoded, i.e., 2D image 10. It does this by computing a likelihood measurement, or likelihood score, for a path defining an ideal image that is the summation of scores for individual transitions in the path.

[0092] FIG. 20 is a flow chart illustrating the sequence of steps implementing the Viterbi decoder of decoding process 330 of the illustrated embodiment. Viterbi image decoding involves path-finding in a three-dimensional decoding lattice, also called a decoding trellis. The decoding lattice is composed of nodes that may be viewed as forming a stack of image planes, one for each node or state of the source model. There is a one-to-one correspondence between the states and paths in the transcription-image network and nodes and paths in the lattice, and corresponding transitions between nodes in the lattice have the same attribute information as transitions between states in the transcription-image network. Thus, in step 334, transcription-image network 870 is first represented in a data structure as the decoding lattice. Next, in box 338, the order in which scores for the nodes in the lattice are to be computed must be determined; this is accomplished by producing a score computation schedule for the recursion, indicating in which order the nodes of the lattice are to be visited and consequently, in which order the node scores are to be computed. Then, the maximum likelihood scores for each node, in the order prescribed by the schedule, are computed, in box 340. For each node, the transition into that node that maximizes the likelihood score is identified and stored. The steps of decoding process 330 have been illustrated as being performed in a particular sequence for purposes of describing the functions that are performed during decoding according to the illustrated implementation; they may be, and usually are, performed contemporaneously in an actual software implementation.

[0093] At the conclusion of decoding, after the likelihood score for the n_F image plane in the decoding lattice has been computed, the most likely complete path found by the Viterbi decoder is retrieved, in box 380, by backtracing through the stored transitions from the final node to the initial node in the decoding lattice to identify the transitions that compose the best path, and to compute the transition image origin positions (x_i, t_i) in 2D image 10 using equations (5) and (6) above. Each transition of the best path defines one transition image origin position. However, not all of these image positions in 2D image 10 are of interest; a filtering step identifies the transitions that indicate estimated glyph image origin positions in 2D image 10 (i.e., the transitions that include as attributes non-null character templates for characters in the glyph sample character set), extracts these image origin positions from all of the identified transition image origin positions, and pairs these image origin positions with the respective character label of the template attribute on each of the identified transitions.

[0094] Decoding provides an estimate of the image origin position of a glyph sample in 2D image 10, but does not provide information about the extent or size of the glyph sample in the image. The image origin positions are considered to be estimates of positions of glyph samples in the input image because decoding may produce imperfect results such as, for example, when an errorful transcription or a noisy 2D image 10 is an input into the training procedure.

[0095] Character template construction process 270 (FIG. 1 11) has been implemented as novel template construction technique 400 in FIG. 13 and produces a set of trained, labeled character templates without prior segmentation of the training data into isolated glyph samples and without identifying bounding boxes for the samples. Template construction technique 400 identifies each glyph sample in the training data using only the x,y coordinate position in 2D image source of glyphs 10 indicating an image origin position and the label identifying the character represented by the glyph sample located at the respective image origin position.

[0096] With reference to FIG. 21, the first step in template construction is to create a template image region, in box 410, for storing each binary character template to be produced from the training data. Each pixel position in each template image region initially indicates a background pixel color value. In principle, the template image region for each character extends over an entire image plane that is unbounded in all directions. However, the support of a template is typically localized to a relatively small region surrounding the template's origin pixel location so that the template image region is selected to be some bounded image region smaller than the entire image plane, but large enough to contain the entire support of a template. FIG. 22 illustrates exemplary template image region 502 which assumes that the support of each template Q_i lies within a rectangle of height H and width W. Template image region 502 will also be referred to as the "canvas" of the template. The shape of the template canvas is fundamentally arbitrary, and is typically selected on the basis of assumptions about the character set for which templates are being trained, and about the samples in the training data.

[0097] The selection of the vertical and horizontal size dimensions of the canvas, i.e. the height H and width W canvas parameters, is made on the basis of two factors that make use of information about the characters in the character set being trained. First, H and W canvas parameters are selected so that the resulting image region created is large enough to entirely contain the support of a single template; in effect, selection of the H and W canvas parameters reflects the decision that pixels outside the canvas are assumed not to be part of the template and are assumed to be the background (white) color. Selection of the H and W canvas parameters is also made so that the resulting image region created in the 2D input image is large enough to entirely contain at least a single glyph sample.

[0098] Template canvas 502 has a local coordinate system associated with it in which x increases to the right, y

increases downward, and the origin 506 of the coordinate system is at $(\chi, -\Psi)$ relative to the lower left corner 508 of canvas 502; thus lower left corner 508 of canvas 502 has coordinates at $(-\chi, \Psi)$ relative to the local coordinate system, where $0 \leq \chi < W$ and $0 \leq \Psi < H$. The canvas rectangle 502 is denoted by C , so that

$$C = [-\chi, -\chi + W - 1] \times [\Psi - H + 1, \Psi] \quad (1)$$

Canvas parameters H , W , χ and Ψ need not be uniform for all templates, and may vary by the particular character template being stored; it is usually more convenient to use the same canvas parameters for each template being trained.

[0099] Each character template includes a pixel position designated as the template's origin that is assumed to lie within canvas 502. The template origin pixel position is illustrated in FIG. 22 as template origin 506. Designation of template origin 506 within canvas rectangle 502 is arbitrary, subject to the constraint that the template to be stored in canvas rectangle 502 must be entirely contained within canvas rectangle 502 when its origin is placed at selected template origin 506. In the illustrated embodiment, satisfactory results have been achieved when template origin 506 is designated to be a pixel position to the left of and below a center pixel position in canvas rectangle 502.

[0100] In FIG. 21, the next step in the template construction procedure of the present invention, in box 430, is to determine a sample image region in the 2D image source of glyphs 10 for each labeled glyph image origin position included in the training data produced as the output of the network merging and decoding processes described above. Template image region 502 is used as a pattern, or guide, in determining two important characteristics of each of these sample image regions: first, the sample image region in 2D image 10 for each labeled glyph image origin position in the training data has vertical and horizontal size dimensions identical to the vertical and horizontal size dimensions (the H and W canvas parameters) of canvas rectangle 502; secondly, the glyph image origin position of the glyph sample is located in the sample image region at a pixel position that is coincident with, or respectively paired with, the pixel position in canvas rectangle 502 designated as template origin position 506. The result of identifying the sample image regions in this manner is to produce a collection of sample image regions in 2D image 10 for each unique character identified by the glyph labels associated with glyph image origin positions in the training data.

[0101] FIG. 23 illustrates three sample image regions 80, 82 and 84 identified for glyph image origin positions 85, 87 and 89 in image region 18 of 2D image 10, each having a glyph label indicating the character "r." Each sample image region has the same height, H , and width, W , of canvas rectangle 502, shown by the H and W designations at the periphery of sample image region 84. Each sample image region has a local coordinate system having its origin aligned at the glyph image origin position, as illustrated in FIG. 23 by origin 85 of representative sample image region 80. Glyph image origin positions 85, 87 and 89 are located at pixel positions in sample image regions 80, 82 and 84 that have x and y displacements from the respective lower left corners of the sample image regions identical to the x and y displacements of template origin 506 from the lower left corner 508 of template canvas rectangle 502.

[0102] Identifying the sample image region for a labeled glyph image origin position can be summarized as follows: if vector $x_i = (x_i, y_i)$ is a glyph origin position within an image of text, the corresponding glyph sample image region is defined to be that portion of the text image within the region defined by $x_i - \chi \leq x < x_i - \chi + W$ and $y_i + \Psi - H < y \leq y_i + \Psi$. That is, the glyph sample image for a glyph position is that portion of the text image within the template canvas, when the template origin is coincident with the glyph origin.

[0103] The term "aligned sample image regions" is introduced to denote the characteristic of each sample image region of the image origin position of the glyph sample being located at a pixel position in the sample image region that has x and y displacements from the lower left corner of the sample image region identical to the x and y displacements of the template image origin 506 from lower left corner 508 of template canvas rectangle 502. The concept of aligned sample image regions is illustrated in FIG. 24 which shows sample image regions 80, 82 and 84 of 2D image 10 from FIG. 23 stacked in layers, one on top of another, above canvas rectangle 502. Respective image origin positions 85, 87 and 89 of sample image regions 80, 82 and 84 are "vertically" aligned with each other, and with template origin position 506, along dotted line axis 88. Alignment of same-sized sample image regions at respective image origin positions in this manner establishes a spatial relationship, or pairing, among each respective pixel location in each of the sample image regions relative to the local coordinate system of the sample image region, and establishes the same spatial relationship, or pairing, between the set of paired pixel locations in the collection of sample image regions and a pixel position in canvas rectangle 502 relative to the template coordinate system. Each set of pixels in aligned sample image regions related in this manner will be referred to as "respectively paired pixels" or "aligned pixels."

[0104] All of the sample image regions identified in 2D image 10 for a particular one of the characters in the character set for which templates are being trained are referred to as a "collection" of sample image regions. In the illustrated implementation, the collection of sample image regions is represented in a separate data structure of sample image regions that are aligned with each other and with template image region 502 at the image origin position. FIG. 25 illustrates data structure 90 that is the collection of sample image regions for the character "a" in the entire scanned newspaper article that is the image represented by 2D image 10. Data structure 90 is presented in FIG. 25 in rows and

columns of concatenated, aligned sample image regions clipped from 2D image 10 according to the pattern provided by canvas rectangle 502; the sample image regions are shown with borders for purposes of illustration.

[0105] With reference again to FIG. 21, the next step in the template construction procedure of the present invention, in box 450, is to assign foreground pixel color values to the pixels in each canvas rectangle 502 for each character template to be produced, on the basis of the pixel color values in the sample image regions. The template construction procedure of the present invention constructs a set of character templates substantially contemporaneously by assigning a color to each pixel in a set of character templates, given a collection of glyph sample images for each of those characters. Unlike conventional practice, a glyph sample image in the technique of the present invention is permitted to contain parts of adjacent glyphs, as illustrated in FIG. 25. The template construction procedure of the present invention effectively determines, while character templates are constructed, which of the foreground pixels within a glyph sample image belong to the central glyph (i.e. the glyph whose origin is aligned with the template origin) and which belong to adjacent glyphs.

[0106] Let $q_t(x)$ denote the color of the pixel at position x of template Q_t , where $t \in B$ is a transition of the Markov image source. A foreground pixel color is represented by a bit value of 1 (one), and a background pixel color by a bit value of 0 (zero). The objective of template construction is to assign a value to $q_t(x)$ for each transition $t \in B$ and for each $x \in C$, given a set of labeled glyph image origin positions (x_i, t_i) , $i = 1 \dots P$.

[0107] Template construction according to the present invention is based on a maximum likelihood (ML) criterion. Template construction using the maximum likelihood criterion involves assigning values to $q_t(x)$ to maximize (25), subject to the template disjointness constraint.

$$L(Z|Q_n) = \sum_{t \in B} \sum_{x \in C} q_t(x) \left[\gamma \sum_{i=1}^{N_t} z(x + x_i^{(t)}) + \beta N_t \right] \quad (2)$$

[0108] If the template disjointness constraint is ignored, template construction using the ML criterion becomes straightforward and consists of separately maximizing each term of the right hand side of equation (2). Since $q_t(x) \in \{0, 1\}$, the ML decision rule is

$$q_t(x) = \begin{cases} 1 & \text{if } S_t(x; Z) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where

$$S_t(x; Z) \equiv \gamma \sum_{i=1}^{N_t} z(x + x_i^{(t)}) + \beta N_t \quad (4)$$

The reason for explicitly noting the dependence of $S_t(x; Z)$ on Z becomes clear shortly from the following discussion. The condition $S_t(x; Z) > 0$ may be written as

$$\frac{1}{N_t} \sum_{i=1}^{N_t} z(x + x_i^{(t)}) > \frac{-\beta}{\gamma} \quad (5)$$

which has the following interpretation: The left hand side of (5) is the fraction of pixels at location x that are black (i.e., foreground pixels) in the collection of aligned sample image regions for Q_t . Thus, $S_t(x; Z)$ may be referred to as an "aligned pixel score" or a "template contribution measurement" at location x for template Q_t . The ML decision rule (3) prescribes that the template pixel at x should be black if the fraction of black pixels at location x in the aligned sample image regions exceeds a threshold. Simply, if the template disjointness constraint is ignored, each ML template may be independently computed by averaging and thresholding the collection of aligned sample image regions for the template, pixel by pixel.

[0109] FIG. 26 shows three templates 94, 96 and 98, selected from a set of templates, constructed from collections

of sample image regions for the characters "e," "a" and "r," respectively, using decision rule (3) without observing template disjointness constraint. The sample image regions used were similar to those in FIG. 25 and were extracted from the scanned image of a newspaper column similar to 2D image 10. It can be seen that templates 94, 96 and 98 clearly include the "correct" template images 93, 95 and 97, aligned at the origin of each canvas rectangle (indicated by the "+" sign). However, it can also be seen that each template canvas includes black pixels that clearly do not belong to the template. These extra black pixels occur in the templates when the averaging and thresholding operations of decision rule (3) are performed on neighboring glyphs in each of the sample image regions in the collection for a template. The extra pixels clearly arise as a result of using sample image regions that contain multiple glyphs, as opposed to a single, isolated glyph. If the sample image regions had contained only the central glyph of interest, e.g. as required in conventional template construction techniques, these extra pixels would be missing.

[0110] Maximizing equation (2), subject to the template disjointness constraint is a computationally difficult problem, in the formal sense of being NP-complete. Rather than use an exponential algorithm to solve the constrained ML template construction problem exactly, the template construction technique of the present invention provides an approximate but effective solution that produces templates that substantially observe the template disjointness constraint. This solution, illustrating the details of box 450 in FIG. 21, is shown in flowchart form in FIG. 27.

[0111] Rather than apply equation (3) independently to each template pixel included in a single template, on a pixel-by-pixel basis, a value of 1 is assigned, in some sequential order, to each template pixel - in any template - for which $S_t(x;Z) > 0$, thereby producing assigned template pixels. After each such assignment, the observed image Z - that is, the sample image regions clipped from the 2D image source of glyph samples in the illustrated embodiment - is modified by setting to zero (0) all aligned sample pixels at locations that are paired with, or coincident with, the newly assigned template pixel. For example, suppose that template pixel $q_s(w) = 1$ has just been assigned. Then the pixels of Z at locations $w + x_i^{(s)}$, $i = 1 \dots N_s$ are set to 0 before the next template pixel assignment is made to a remaining unassigned template pixel. The effect of setting sample pixels in the observed image to zero after a coincident template assignment has been made, which may be called "clearing pixels of Z," is to reduce the value of $S_t(x;Z)$, for subsequent computations of $S_t(x;Z)$, for overlapping template pixels that have not yet been set to 1, thereby decreasing the likelihood that the overlapping pixels will be set to 1 subsequently. The sequential assignment continues as long as $S_t(x;Z) > 0$ for some unassigned template pixel. The net result of the template construction technique of the present invention is to produce the entire set of trained character templates contemporaneously, with no one template being complete until no positive $S_t(x;Z)$ remains.

[0112] With reference to FIG. 27, after initializing pixel scores, or template contribution measurements, $S_t(x;Z)$, associated with each pixel position in each template canvas, in box 452, to some value greater than zero, $S_t(x;Z)$ is computed, in box 456, for each unassigned template pixel in each template having a currently positive pixel score, using the respectively paired aligned sample pixel positions in the collection of aligned sample image regions for that template. The pixel scores are then tested in box 460, and if any one of the computed pixel scores is greater than zero, the procedure proceeds to box 470 where the template pixel, in any template, having the highest positive pixel score is selected, and a foreground color value is assigned to that selected template pixel. The color values of the aligned pixels in the collection of aligned sample image regions paired with the selected template pixel are then set to zero (i.e., the background color value) in box 480. Then processing returns to box 456 where pixel scores are again computed for remaining unassigned template pixels.

[0113] FIG. 28 shows the results of applying the template pixel color assignment algorithm to the same glyph sample image data used to generate the templates shown in FIG. 26. The set of templates 510 in FIG. 28 are arranged in the order "space", lowercase letters, uppercase letters, numerals and punctuation. If a character does not occur in the input image its template is given as a solid black square. Compared with FIG. 26, templates 510 in FIG. 28 contain significantly fewer extra black pixels, reflecting the effect of the "Z pixel clearing" step of the algorithm. In particular, templates 516, 514 and 518 representing characters "e," "a" and "r," respectively, have been called out for purposes of comparing them to templates 94, 96 and 98 in FIG. 26. As noted in (27), computation of the pixel scores requires use of the factors γ and β , where $\gamma > 0$ and $\beta < 0$. In the illustrated embodiment that produced the templates shown in FIG. 26, the values used for these factors were 2.237 and - 1.629, respectively, corresponding to channel noise parameters $\alpha_0 = .9$ and $\alpha_1 = .51$.

[0114] Each transition t_i between nodes in the finite state image model network has the associated 4-tuple of attributes, shown in FIG. 14 in the order $[a_i] (\Delta_i), m_i, Q_i$. When a template, Q_i , is associated with a transition t_i , such as those illustrated with the F transitions t_i , in FIG. 14, the horizontal displacement, Δ_i , associated with that transition is the character set width, W_{t_i} , of the template. The character set width is the vector displacement from the glyph origin position to the point at which the origin of the next glyph is normally placed when imaging consecutive characters of a word. The character set width is one of the font metrics needed to completely describe a character template that is modeled by the sidebearing model of letterform shape description and positioning. Therefore, in addition to constructing the character templates according to template construction procedure 400, it is also necessary to determine the character set widths for the constructed templates.

[0115] The character set width of each binary template is determined using the collection of sample image regions identified for that template. Because identifying glyph image origin positions of glyph samples in the 2D input image is a process of estimation, it is likely that at least some of the identified samples will have inaccurate image origin positions identified. However, the set width of each glyph sample included in a sample image region can be computed from knowledge of the image origin position of the next adjacent glyph sample in the 2D image. Therefore, computing a set width for a template includes computing the set width for each sample identified for that template using the collection of sample image regions and the displacement from each image origin position in each sample to the image origin position of the next adjacent glyph in the 2D image. The collection of computed set widths for the glyph samples is then used to arrive at a set width for the template; for example, the mean or median set width value for all of the samples may be determined to be the set width for the template. Or the minimum set width computed using the samples may be used as the template's set width.

[0116] FIG. 13 shows this step of determining the character set widths as box 490, following template construction procedure 400. However, as just described, in the illustrated implementation the set width of each template is determined using the collections of sample image regions, and not from using the constructed templates. Therefore, the determination of character set widths is not dependent on the completion of template construction, and may take place at any point after the decoding and backtracing steps produces the labeled glyph image origin positions for the glyph samples in the 2D input image. In addition, FIG. 13 shows this step as being included in an iterative processing loop that repeats the decoding, backtracing and template construction steps 330, 380 and 400, respectively. The preferred method for computing the set width is to determine the minimum set width from the set widths computed for the collection of sample image regions, and then to take a percentage of that minimum, for example 90 per cent, as the set width for the template, in order to ensure that the set widths used for character positioning during subsequent iterations of the decoding process is always going to be less than the actual set widths used for positioning the glyphs in the input 2D input image.

[0117] The illustrated embodiment of character template training technique 200 is fundamentally an iterative process because, as has been previously noted, image decoding of an observed 2D image using 2D image source models of the type illustrated in FIGS. 9, 14 and 19 assumes the use of an initial set of character templates. When, as is typical in the situation of training, an initial set of templates is not available, the illustrated embodiment includes processing, prior to decoding step 330, for generating a character template data structure indicating an initial set of character templates for use during decoding. Each template in this initial set of character templates may have any arbitrary pixel content that is of practical use by decoding process 330, and in the illustrated embodiment each template has the arbitrary initial pixel content of a solid black rectangular image, and has no specific pixel information about the character it represents. Given such an initial set of templates of rectangular black images, decoding and backtracing steps 330 and 380 respectively are likely to produce improved estimates of labeled glyph image origin positions of glyph samples in 2D image 10 with each subsequent iteration, using the character templates constructed in the previous iteration.

[0118] The stopping condition used to control the completion of character template construction may be heuristically determined, or may be a function of one or more processing parameters. In the illustrated embodiment, the stopping condition is a fixed number of iterations that has proven from experience to be the number of iterations that produces the highest quality templates and after which subsequent observable improvements to the templates are of little or no significance. The stopping condition may also be based on a threshold related to the maximum likelihood score computed during decoding.

[0119] The flexibility of having both the 2D image source model and the transcription represented as formal models and of having the 2D image source model represented as an explicit input to the training procedure expands the concept of what is traditionally thought of as the type of transcription suitable for training templates - namely, the literal transcription - to include a wide variety of other message strings. For example, a situation may arise in which the 2D input image source of glyph samples used for training is always one of set of specific documents having known, fixed transcriptions. Transcription networks, or transcription-image networks, modeling these predetermined transcriptions may be produced and stored in advance of training for each of these specific transcriptions, and what the user enters as a "transcription" is actually a name identifying the specific transcription to be associated with the 2D image source of glyph samples to be used in training.

[0120] In another example, an available data structure that includes markup labels, or tags, indicating the logical structure and formatting information for the character codes in a 2D image may also be used as the input transcription, without the user having to manually strip out the tags and transform the data structure into a literal transcription. If these markup data structures are available in document depositories along with their document images, the training technique of the present invention makes it possible to train character templates using any such document image and its associated tag transcription.

[0121] The use of tagged transcriptions in the template training technique of the present invention requires no functional modifications to the image models described thus far because accommodating tags essentially involves the handling of the message strings in the image source network, and the general form of Markov image source model

given in FIG. 9 allows independent specification of the message string and template for each transition. Moreover, either or both of the message and template may be omitted from a transition attribute set. Thus, tags may be accommodated without modification to the modeling framework, as will be illustrated below.

[0122] FIG. 29 shows a line image source model 770 that images simple text lines that contain subscripts. The transition probabilities have been omitted from this model. The line images defined by model 770 consist entirely of glyphs of the character "a" with various amounts of intercharacter space. The states and their transitions in model 770 show that glyphs can be placed either on the baseline or 5 pixels below, to simulate subscripts. From the transition attributes shown in model 770, it can be seen that state n_2 and its self-transitions 777 and 778 image consecutive templates of the character "a" that are aligned on the main baseline, and generate message strings of "a." It can also be seen from the displacement vector on transition 778 that the set width of the "a" in the horizontal x direction is given as 25. Model 770 permits the imaging of glyphs on a subscript baseline by the transition from state n_1 to state n_4 which shows, as transition attributes, a positive y displacement of 5 that moves the current imaging position down in the image plane, and message string 772, "{", but no template. State n_4 and its self-transitions 779 and 780 image consecutive templates of the character "a" that are aligned on a subscript baseline, and also generate message strings of "a." The transition from state n_4 to state n_3 returns the current imaging position to the main baseline, as shown by displacement 776; this transition also has message string attribute 774, "}", and no template. Because of the transition from n_3 to n_1 , a line may contain any number of subscript strings alternated with strings on the main baseline. (Because model 770 models a set of images that includes text that is imaged below a main text baseline, all branch displacements are specified as 2-dimensional vectors. However, it is simple to verify that all complete paths through model 770 have zero y displacement, i.e., $\Delta y_\pi = 0$ if π is a complete path. Thus, the model satisfies the defining condition to be a line model, i.e., every complete path has the same y displacement.)

[0123] FIG. 30 shows several examples of line images 712, 714 and 716 that are included in the set of line images modeled by line image source model 770. For purposes of illustration, the vertical displacement of the subscripts in FIG. 30 is greatly exaggerated, and the dashed lines, such as line 718, are shown to illustrate the main text baseline. In its decoding mode, model 770 decodes line images 712, 714 and 716 as the message strings (transcriptions) "aaa{aa}a", "a{a}aa{a}a" and "a{a}a", respectively. Or conversely, line images 712, 714 and 716 could be viewed as line images that model 770 generates in its image synthesis mode, given the input message strings "aaa{aa}a", "a{a}aa{a}a" and "a{a}a", respectively. In either event, the actual message strings "{" and "}" cannot be visually paired with glyphs in any of the line images 712, 714 and 716 in FIG. 30, a fact that is supported by the missing templates on the transitions from state n_1 to state n_4 and from state n_4 to state n_3 . When encountered in model 770, message strings "{" and "}" indicate a perceptible change in the imaging of one or more glyphs in an image generated by the model. Message strings "{" and "}" thus function as tags that mark text between them as having a formatting or logical change in imaging from text that precedes or follows them; in this case, they mark text that is to be interpreted and typeset as a subscript.

[0124] The merging of an image source model that accommodates tag message strings with a tagged transcription network proceeds in the same manner as previously described for merging 2D networks and line networks. The functional properties of the merged tag transcription image network are the same as those presented previously in the context of the 2D implementation (referred to there as network properties (a) and (b)). Network merging produces a modified image source model that is constrained to generate only transcriptions from the set of transcriptions defined by the tag transcription network. The merging process follows the three steps for adding transitions to the states in the line image model that are described above.

[0125] Using a tagged transcription as the input source of glyph labels for the training data produced for the template training procedure is entirely handled by how the image and transcription models are defined and merged, and requires no modifications to the decoding process or to glyph image origin position extraction from the best path; the remainder of the template training procedure proceeds as previously described, using tag transcription image network to provide the glyph image origin positions of the glyphs included in an input line image to the template construction procedure.

Claims

1. A method for training a set (20) of character templates (22,24,26) for use in a recognition system; each character template representing a respective one of a plurality of characters in a character set and being identified by a character label indicating the respective character; the method being for operating a machine including a processor and a memory device connected thereto; the memory device storing instruction data which the processor executes; the method comprising the steps of:

receiving a two-dimensional source image (10,50) including a plurality of glyph samples (12,14); each glyph sample being an image instance of a respective one of the plurality of characters in the character set; the

source image having a vertical dimension size larger than the height of a single line of glyph samples;

receiving a transcription data structure (40,60,70) being associated with the source image and including an ordered arrangement of transcription labels (42,44,46,47,48,62,64,66,68,72);

determining (220) a glyph pixel position of each glyph sample;

producing (250) for each glyph sample a glyph label indicating a respective one of the characters in the character set and pairing (250) same with the respective glyph pixel position; and

producing (270,400) the set of character templates using the pairs of glyph pixel positions and glyph labels and using the glyph samples at the glyph pixel positions as training data samples for a respective one of the character templates,

characterized in that

the memory device further stores a two-dimensional image source model (820,830,770) for modeling as a grammar a spatial image structure of a set of two-dimensional images; the image source model including spatial positioning data for modeling spatial positioning of a plurality of glyphs included in an image and mapping data for mapping a respective one of the glyphs to a glyph label indicating the character in the character set; the source image being one image of the set of images;

the step of determining the glyph pixel positions uses the spatial positioning data; and

the step of producing and pairing the glyph labels uses the mapping data and the transcription data structure.

2. The method according to claim 1, further **characterized in that** each glyph pixel position is a single origin position of the respective glyph sample and the step of producing (270,400) the set of character templates includes the steps of:

determining (430) regions in the source image, each region including a plurality of sample pixel positions and being large enough such that two of the sample pixel positions are the origin positions of two glyph samples; and

assigning (450) pixel color values to template pixel positions included in respective ones of the character templates using pixel color values indicated by the sample pixel positions on the basis of criteria that determine which sample pixel positions in which regions are used to determine a pixel color value for a template pixel position such that all character templates observe a character template model according to which two adjacent character templates being displaced from each other by a character set width may have overlapping bounding boxes but substantially nonoverlapping foreground pixels, each bounding box entirely containing one character template.

3. The method according to claim 1 or claim 2, further **characterized in that** the transcription data structure is a tag transcription data structure including at least one nonliteral tag transcription label indicating at least one character code representing a character with which no glyph sample in the source image can be paired by visual inspection thereof; the at least one character code indicating markup information about the source image; the markup information, when interpreted by a document processing operation, being capable of producing at least one display feature included in the source image perceptible as a visual formatting characteristic; and wherein the step of producing (250) the glyph label uses the spatial positioning data about the plurality of glyph samples to identify at least one glyph sample related to the tag data structure, and wherein the step of pairing (250) uses the tag data structure for pairing.

4. A method for training a set of character templates for use in a recognition system; each character template representing a respective one of a plurality of characters in a character set; the method being for operating a machine including a processor and a memory device connected thereto; the memory device storing instruction data which the processor executes; the method comprising the steps of:

receiving and storing (290) a two-dimensional source image including a plurality of glyph samples; each glyph sample being an image instance of a respective one of the plurality of characters in the character set; the

source image having a vertical dimension size larger than the height of a single line of glyph samples;

receiving and storing (292) a finite-state transcription network (800,850,852) being associated with the source image and including an ordered arrangement of transcription labels as at least one transcription path through the transcription network;

decoding (330) the source image;

obtaining (380) for each glyph sample in the source image a position and a glyph label paired therewith; and

producing (400) the set of character templates using the source image and the pairs of positions and glyph labels and using the glyph samples identified by the glyph labels for training the character templates,

characterized in that

the memory further stores a two-dimensional stochastic finite state image source network (830) for modeling as a grammar a spatial image structure of a set of two-dimensional images, each including a plurality of glyphs; a first image of the set of images being modeled as at least one path through the image source network that indicates an ideal image consistent with the spatial image structure of the first image; the at least one path indicating path data items associated therewith; the path data items indicating positions and glyph labels paired therewith of respective ones of the plurality of glyphs included in the first image; the source image being one image of the set of images;

the method further comprises a step of merging (300) the image source network with the transcription network to produce a transcription-image network (870,872) wherein, when the transcription is associated with the first image, the transcription-image network models the first image as at least one complete transcription-image path through the transcription-image network that indicates an ideal image consistent with the spatial image structure of the first image; the at least one complete transcription-image path indicating the path data items and a sequence of message strings consistent with the ordered arrangement of the transcription labels;

the step of decoding uses the transcription-image network to produce at least one complete transcription-image path indicating an ideal image consistent with the spatial image structure of the source image; and

the step of obtaining uses the path data items associated with the at least one complete transcription-image path;

5. The method according to claim 4, further characterized in that

each character template is based on a sidebearing model of character image positioning of two adjacent character images being displaced from each other by a character set width; the sidebearing model requiring that, when two rectangular bounding boxes each containing one of the two character images overlap, the character images have substantially nonoverlapping foreground pixels; and wherein the step of producing (400) the set of character templates includes the step of determining (490), for each character template, a character set width thereof using the pairs of positions and glyph labels; or that

each of the character templates is based on a segmentation-based model of character image positioning of two adjacent character images, each of which being capable of being entirely contained in one of two substantially nonoverlapping rectangular bounding boxes.

6. The method according to claim 4, further characterized in that

the transcription network includes at least one pair of transcription labels indicating at least first and second alternate message strings for a single image portion of the source image; the transcription network including a series of transcription nodes and a sequence of transitions between pairs of the nodes; each transition having a transcription label associated therewith; the transcription network modeling the at least one pair of transcription labels as at least first and second alternate sequences of transitions between one of the pairs of the nodes; each one of the at least first and second alternate sequences of transitions having one of the at least first and second alternate message strings associated therewith; or that

the transcription network indicates a tag transcription including at least one nonliteral tag transcription label indicating at least one character code representing a character with which a respective glyph in the source image cannot be paired by visual inspection thereof; the at least one character code indicating markup information about the source image; the markup information, when interpreted by a document processing operation, being capable of producing at least one display feature included in the source image perceptible as a visual formatting characteristic; and the step of merging (300) including the step of designating the transcription labels as message strings included in the transcription-image network; the tag transcription label being designated as a message string such that the transcription-image network models a relationship between the tag transcription label and at least one glyph; or that

the at least one complete transcription-image path is defined by a sequence of transitions between nodes included in the transcription-image network; the path data items associated therewith being associated with each transition and including a message string, a character template, and an image displacement; and the step of obtaining (380) includes the step of determining, for respective transitions having non-null character templates associated therewith, the positions of respective glyphs by using the image displacements associated with the respective transitions and further, determining the glyph label paired with each respective position using a character label indicated by the non-null character template associated with the respective transition; the character label indicating a character in the character set.

7. The method according to claim 4, further **characterized in that** the step of decoding (330) includes the steps of:

performing (334,338,340) a dynamic programming based decoding operation to compute an optimum score at each of a plurality of lattice nodes in a decoding lattice representing the transcription-image network; said decoding operation producing for each lattice node an optimizing transition identification data item as a result of computing the optimum score, storing the item and indicating one of a plurality of possible transitions into a respective one of the lattice nodes that optimizes the score for the respective lattice node; and

backtracing (380) to retrieve a sequence of transitions indicating a decoding lattice path; said backtracing operation starting with a final lattice node and ending with a first lattice node in the decoding lattice path; the sequence of transitions being retrieved using the data item; the decoding lattice path indicating the at least one complete transcription-image path through the transcription-image network.

8. A machine for training a set of character templates for use in a recognition operation, comprising:

a first signal source for providing a first image;
 image input circuitry for receiving said image from the first signal source;
 a second signal source for providing non-image data;
 input circuitry for receiving said non-image data from the second signal source;
 a memory for storing data including instructions which a processor can execute; and
 the processor for receiving said image from the image input circuitry and for receiving said non-image data from the input circuitry and for accessing the data stored in the memory; the processor being arranged for, in executing the instructions,
 receiving from the image input circuitry a two-dimensional source image (10,50) including a plurality of glyph samples; the source image having a vertical dimension size larger than a single line of glyph samples; each glyph sample being an image instance of a respective one of a plurality of characters in the character set; the set of character templates representing respective ones of the plurality of characters in the character set;
 receiving from the input circuitry a transcription data structure (40,60,70) being associated with the source image and including an ordered arrangement of transcription labels (42,44,46,47,48,62,64,66,68,72);
 determining a glyph pixel position of each glyph sample;
 producing a glyph label paired with each respective glyph pixel position; and
 producing the set of character templates using the source image and the glyph pixel positions and the respectively paired glyph labels; each character template being identified by a character label; each pair of glyph pixel positions and glyph labels identifying a training data sample for a respective one of the character templates when the respectively paired glyph label matches the respective character label,

characterized in that

the memory further stores a two-dimensional image source model (820,830,770), for modeling as a grammar

a spatial image structure of a set of two-dimensional images; the image source model including spatial positioning data for a plurality of glyphs occurring in a first image included in the set of images; the image source model indicating mapping data for mapping a respective glyph sample occurring in the first image to a glyph label indicating a character in a character set; the source image being included in the set of images; and

the processor is further arranged to use the spatial positioning data for determining the glyph pixel positions and to use the mapping data and the ordered arrangement of transcription labels for producing the paired glyph labels.

9. The machine according to claim 8, further characterized in that

the first signal source is a scanning device; the image input circuitry is scanning circuitry capable of receiving an image by a scanning operation performed on a physical document showing marks indicating the image; the processor is arranged for receiving the source image from the scanning circuitry; and/or that

the second signal source includes a user input device for receiving signals from a user; the input circuitry being connected for receiving the signals from the user input device; and the processor is arranged for receiving the transcription data structure from the input circuitry; and/or that

the image source model is represented as a stochastic finite state network indicating a regular grammar and representing the mapping data mapping a respective glyph pixel position in the source image to a glyph label as a path through the image source network; the path indicating path data items associated therewith and being accessible by the processor; the path data items indicating positions and respectively paired glyph labels of respective glyphs included in the source image; and

the transcription data structure is represented as a finite state network (800,850,852) modeling a transcription as a transcription path through the transcription network indicating the ordered arrangement of transcription labels; and

the processor is arranged for, in determining the glyph pixel positions and producing respectively paired glyph labels,

merging the image source network with the transcription network to produce a transcription-image network (870,872) indicating modified mapping data for mapping a respective transcription label to a glyph pixel position of a respective glyph occurring in the source image; the modified mapping data being represented as at least one complete transcription-image path through the transcription-image network; the transcription-image path indicating the path data items;

decoding the source image using the transcription-image network to produce the transcription-image path; and obtaining the glyph pixel position of each glyph sample and the glyph label paired therewith using the path data items indicated by the transcription-image path.

10. The machine according to claim 8, further characterized in that

the transcription data structure is a tag transcription data structure including at least one nonliteral tag transcription label indicating at least one character code representing a character with which no glyph sample can be paired by visual inspection thereof; the character code indicated by the tag transcription label indicating markup information about the source image; the markup information, when interpreted by a document processing operation, producing at least one display feature included in the source image perceptible as a visual formatting characteristic; and

the processor is arranged for, in producing the paired glyph label, using the spatial positioning data to identify at least one glyph sample related to the tag transcription label and using the tag transcription label.

Patentansprüche

1. Verfahren zum Trainieren eines Satzes (20) von Zeichenschablonen (22, 24, 26) zur Verwendung in einem Erkennungssystem; wobei jede Zeichenschablone ein jeweiliges Zeichen einer Vielzahl von Zeichen in einem Zeichensatz repräsentiert und durch eine Zeichenlabel identifiziert wird, das das jeweilige Zeichen angibt; wobei das Verfahren zum Betreiben einer Maschine ist, die einen Prozessor und eine damit verbundene Speichereinrichtung enthält, wobei die Speichereinrichtung Instruktionsdaten speichert, die der Prozessor ausführt; wobei das Verfahren die folgenden Schritte umfasst:

Empfangen eines zweidimensionalen Quellenbildes (10, 50), das eine Vielzahl von Glyphproben (12, 14) enthält; wobei jede Glyphprobe ein Bildbeispiel eines jeweiligen Zeichens der Vielzahl von Zeichen in dem Zeichensatz ist; wobei das Quellenbild eine vertikale Dimensionsgröße hat, die größer als die Höhe einer einzelnen Zeile von Glyphproben ist;

Empfangen einer Transkriptionsdatenstruktur (40, 60, 70), die dem Quellenbild zugeordnet ist und eine geordnete Anordnung von Transkriptionslabeln (42, 44, 46, 47, 48, 62, 64, 66, 68, 72) enthält;

Bestimmen (220) einer Glyphpixelposition jeder Glyphprobe;

Erzeugen (250) eines Glyphlabels, das ein jeweiliges Zeichen in dem Zeichensatz angibt, für jede Glyphprobe und Paaren (250) derselben mit der jeweiligen Glyphpixelposition; und

Erzeugen (270, 400) des Satzes von Zeichenschablonen unter Verwendung der Paare von Glyphpixelpositionen und Glyphlabeln und unter Verwendung der Glyphproben an den Glyphpixelpositionen als Trainingsdatenproben für eine jeweilige Zeichenschablone,

dadurch gekennzeichnet, dass

die Speichereinrichtung weiterhin ein zweidimensionales Bildquellenmodell (820, 830, 770) zum Modellieren einer räumlichen Bildstruktur eines Satzes zweidimensionaler Bilder als eine Grammatik speichert; wobei das Bildquellenmodell räumliche Positionierungsdaten zum Modellieren einer räumlichen Positionierung einer Vielzahl von in einem Bild enthaltenen Glyphs und Mappingdaten zum Abbilden eines jeweiligen Glyphs auf ein das Zeichen in dem Zeichensatz angegebendes Glyphlabel enthält; wobei das Quellenbild ein Bild des Satzes von Bildern ist;

der Schritt des Bestimmens der Glyphpixelpositionen die räumlichen Positionierungsdaten verwendet; und

der Schritt des Erzeugens und Paarens der Glyphlabel die Mappingdaten und die Transkriptionsdatenstruktur verwendet.

2. Verfahren nach Anspruch 1, weiterhin **dadurch gekennzeichnet**, dass jede Glyphpixelposition eine einzelne Ursprungsposition der jeweiligen Glyphprobe ist und der Schritt des Erzeugens (270, 400) des Satzes von Zeichenschablonen die folgenden Schritte enthält:

Bestimmen (430) von Bereichen in dem Quellenbild, wobei jeder Bereich eine Vielzahl von Probenpixelpositionen enthält und groß genug ist, so dass zwei der Probenpixelpositionen die Ursprungspositionen zweier Glyphproben sind; und

Zuweisen (450) von Pixelfarbwerten zu Schablonenpixelpositionen, die in jeweiligen Zeichenschablonen enthalten sind, unter Verwendung von durch die Probenpixelpositionen angegebenen Pixelfarbwerten, auf der Grundlage von Kriterien, die bestimmen, welche Probenpixelpositionen in welchen Bereichen verwendet werden, um einen Pixelfarbwert für eine Schablonenpixelposition zu bestimmen, so dass alle Zeichenschablonen ein Zeichenschablonenmodell beachten, wonach zwei benachbarte Zeichenschablonen, die voneinander um eine Zeichensatzbreite verschoben sind, überlappende Bounding-Boxen aber im Wesentlichen nichtüberlappende Vordergrundpixel haben können, wobei jede Bounding-Box eine Zeichenschablone vollständig enthält.

3. Verfahren nach Anspruch 1 oder 2, weiterhin **dadurch gekennzeichnet**, dass die Transkriptionsdatenstruktur eine Tag-Transkriptionsdatenstruktur ist, die wenigstens ein nichtwörtliches Tag-Transkriptionslabel enthält, das wenigstens einen Zeichenkode angibt, der ein Zeichen repräsentiert, mit dem keine Glyphprobe in dem Quellenbild durch visuelles Betrachten desselben gepaart werden kann; wobei der wenigstens eine Zeichenkode Markup-Informationen über das Quellenbild angibt; wobei die Markup-Informationen bei ihrer Interpretierung durch einen Dokumentenverarbeitungsvorgang wenigstens ein in dem Quellenbild enthaltenes Displaymerkmal erzeugen können, das als visuelle Formatierungscharakteristik wahrnehmbar ist; und wobei der Schritt des Erzeugens (250) des Glyphlabels die räumlichen Positionierungsdaten über die Vielzahl von Glyphproben verwendet, um wenigstens eine Glyphprobe zu identifizieren, die mit der Tag-Datenstruktur in Beziehung steht, und wobei der Schritt des Paarens (250) die Tag-Datenstruktur zum Paaren verwendet.

4. Verfahren zum Trainieren eines Satzes von Zeichenschablonen zur Verwendung in einem Erkennungssystem; wobei jede Zeichenschablone ein jeweiliges Zeichen einer Vielzahl von Zeichen in einem Zeichensatz repräsentiert; wobei das Verfahren zum Betreiben einer Maschine ist, die einen Prozessor und eine damit verbundene Speichereinrichtung enthält; wobei die Speichereinrichtung Instruktionsdaten speichert, die der Prozessor ausführt; wobei das Verfahren die folgenden Schritte umfasst:

Empfangen und Speichern (290) eines zweidimensionalen Quellenbildes, das eine Vielzahl von Glyphproben enthält; wobei jede Glyphprobe ein Bildbeispiel eines jeweiligen Zeichens der Vielzahl von Zeichen in dem Zeichensatz ist; wobei das Quellenbild eine vertikale Dimensionsgröße hat, die größer als die Höhe einer einzelnen Zeile von Glyphproben ist;

Empfangen und Speichern (292) eines Finitzustandstranskriptionsnetzwerks (800, 850, 852), das dem Quellenbild zugeordnet ist und eine geordnete Anordnung von Transkriptionslabeln als wenigstens einen Transkriptionspfad durch das Transkriptionsnetzwerk enthält;

Dekodieren (330) des Quellenbildes;

Erlangen (380) einer Position und eines damit gepaarten Glyphlabels für jede Glyphprobe in dem Quellenbild; und

Erzeugen (400) des Satzes von Zeichenschablonen unter Verwendung des Quellenbildes und der Paare von Positionen und Glyphlabeln und unter Verwendung der Glyphproben, die durch die Glyphlabel identifiziert werden, zum Training der Zeichenschablonen,

dadurch gekennzeichnet, dass

der Speicher weiterhin ein zweidimensionales stochastisches Finitzustandsbildquellennetzwerk (830) zum Modellieren einer räumlichen Bildstruktur eines Satzes zweidimensionaler Bilder als eine Grammatik speichert, wobei jedes Bild eine Vielzahl von Glyphs enthält; wobei ein erstes Bild des Satzes von Bildern als wenigstens ein Pfad durch das Bildquellennetzwerk modelliert wird, der ein ideales Bild angibt, das zu der räumlichen Bildstruktur des ersten Bildes konsistent ist; wobei der wenigstens eine Pfad Pfaddatenelemente angibt, die ihm zugeordnet sind; wobei die Pfaddatenelemente Positionen und damit gepaarte Glyphlabel jeweiliger Glyphs der Vielzahl von in dem ersten Bild enthaltenen Glyphs angeben; wobei das Quellenbild ein Bild des Satzes von Bildern ist;

das Verfahren weiterhin einen Schritt des Verschmelzens (300) des Bildquellennetzwerkes mit dem Transkriptionsnetzwerk umfasst, um ein Transkriptionsbildnetzwerk (870, 872) zu erzeugen, wobei, wenn die Transkription dem ersten Bild zugeordnet ist, das Transkriptionsbildnetzwerk das erste Bild als wenigstens einen vollständigen Transkriptionsbildpfad durch das Transkriptionsbildnetzwerk modelliert, der ein ideales Bild angibt, das zu der räumlichen Bildstruktur des ersten Bildes konsistent ist; wobei der wenigstens eine vollständige Transkriptionsbildpfad die Pfaddatenelemente und eine Abfolge von Nachrichtenstrings angibt, die zu der geordneten Anordnung der Transkriptionslabel konsistent ist;

der Schritt des Decodierens das Transkriptionsbildnetzwerk verwendet, um wenigstens einen vollständigen Transkriptionsbildpfad zu erzeugen, der ein ideales Bild angibt, das zu der räumlichen Bildstruktur des Quellenbildes konsistent ist; und

der Schritt des Erlangens die Pfaddatenelemente verwendet, die dem wenigstens einen vollständigen Transkriptionsbildpfad zugeordnet sind.

5. Verfahren nach Anspruch 4, weiterhin **dadurch gekennzeichnet, dass**

jede Zeichenschablone auf einem Sidebearing-Modell der Zeichenbildpositionierung zweier benachbarter Zeichenbilder basiert, die voneinander um eine Zeichensatzbreite verschoben sind; wobei das Sidebearing-Modell erfordert, dass die Zeichenbilder im Wesentlichen nichtüberlappende Vordergrundpixel haben, wenn zwei rechteckige Bounding-Boxen, die jeweils eines von zwei Zeichenbildern enthalten, überlappen; und wobei der Schritt des Erzeugens (400) des Satzes von Zeichenschablonen den Schritt des Bestimmens (490), für jede Zeichenschablone, einer Zeichensatzbreite derselben unter Verwendung der Paare von Positionen und Gly-

phlabeln enthält; oder dass

jede der Zeichenschablonen auf einem segmentierungsbasierten Modell der Zeichenbildpositionierung zweier benachbarter Zeichenbilder basiert, von denen jedes vollständig in einer von zwei im Wesentlichen nichtüberlappenden rechteckigen Bounding-Boxen enthalten sein kann.

6. Verfahren nach Anspruch 4, weiterhin **dadurch gekennzeichnet**, dass

das Transkriptionsnetzwerk wenigstens ein Paar von Transkriptionslabeln enthält, die wenigstens erste und zweite abwechselnde Nachrichtenstrings für einen einzelnen Bildteil des Quellenbildes angeben; wobei das Transkriptionsnetzwerk eine Reihe von Transkriptionsknoten und eine Abfolge von Übergängen zwischen Paaren der Knoten enthält; wobei jeder Übergang ein ihm zugeordnetes Transkriptionslabel hat; wobei das Transkriptionsnetzwerk das wenigstens eine Paar von Transkriptionslabeln als wenigstens erste und zweite abwechselnde Abfolgen von Übergängen zwischen einem der Paare der Knoten modelliert; wobei jede der wenigstens ersten und zweiten abwechselnden Sequenzen von Übergängen einen der wenigstens ersten und zweiten abwechselnden, ihr zugeordneten Nachrichtenstrings hat; oder dass

das Transkriptionsnetzwerk eine Tag-Transkription angibt, die wenigstens ein nichtwörtliches Tag-Transkriptionslabel enthält, das wenigstens einen Zeichenkode angibt, der ein Zeichen repräsentiert, mit dem ein jeweiliges Glyph in dem Quellenbild nicht durch visuelle Betrachtung desselben gepaart werden kann; wobei der wenigstens eine Zeichenkode Markup-Informationen über das Quellenbild angibt; wobei die Markup-Informationen bei ihrer Interpretation durch einen Dokumentenverarbeitungsvorgang wenigstens ein in dem Quellenbild enthaltenes Displaymerkmal erzeugen können, das als eine visuelle Formatierungscharakteristik wahrnehmbar ist; und wobei der Schritt des Verschmelzens (300) den Schritt des Bezeichnens der Transkriptionslabel als Nachrichtenstrings enthält, die in dem Transkriptionsbildnetzwerk enthalten sind; wobei das Tag-Transkriptionslabel als ein Nachrichtenstring bezeichnet ist, so dass das Transkriptionsbildnetzwerk eine Beziehung zwischen dem Tag-Transkriptionslabel und wenigstens einem Glyph modelliert; oder dass der wenigstens eine vollständige Transkriptionsbildpfad durch eine Abfolge von Übergängen zwischen Knoten, die in dem Transkriptionsbildnetzwerk enthalten sind, definiert ist; wobei die ihm zugeordneten Pfaddatenelemente jedem Übergang zugeordnet sind und einen Nachrichtenstring, eine Zeichenschablone und eine Bildverschiebung enthalten; und wobei der Schritt des Erlangens (380) den Schritt des Bestimmens, für jeweilige Übergänge mit ihnen zugeordneten, von Null verschiedenen Zeichenschablonen, der Positionen jeweiliger Glyphs durch die Verwendung der Bildverschiebungen, die den jeweiligen Übergängen zugeordnet sind, und weiterhin des Bestimmens des mit jeder jeweiligen Position gepaarten Glyphlabels unter Verwendung eines Zeichenlabels enthält, das durch die von Null verschiedene Zeichenschablone angegeben ist, die dem jeweiligen Übergang zugeordnet ist; wobei das Zeichenlabel ein Zeichen in dem Zeichensatz angibt.

7. Verfahren nach Anspruch 4, weiterhin **dadurch gekennzeichnet**, dass der Schritt des Dekodierens (330) die folgenden Schritte enthält:

Durchführen (334, 338, 340) eines auf dynamischer Programmierung basierenden Dekodierungsvorgangs, um an jedem einer Vielzahl von Gitterknoten in einem Dekodierungsgitter, das das Transkriptionsnetzwerk repräsentiert, einen optimalen Score zu berechnen; wobei der Dekodierungsvorgang für jeden Gitterknoten ein optimierendes Übergangsidentifikationsdatenelement als Ergebnis der Berechnung des optimalen Scores erzeugt, das Element speichert und einen einer Vielzahl möglicher Übergänge in einen jeweiligen Gitterknoten angibt, der den Score für den jeweiligen Gitterknoten optimiert; und

Zurückverfolgen (380), um eine Abfolge von Übergängen wiederzugewinnen, die einen Dekodierungsgitterpfad angibt; wobei der Zurückverfolgungsvorgang mit einem Endgitterknoten beginnt und mit einem ersten Gitterknoten in dem Dekodierungsgitterpfad endet; wobei die Abfolge von Übergängen unter Verwendung des Datenelements wiedergewonnen wird; wobei der Dekodierungsgitterpfad den wenigstens einen vollständigen Transkriptionsbildpfad durch das Transkriptionsbildnetzwerk angibt.

8. Maschine zum Trainieren eines Satzes von Zeichenschablonen zur Verwendung in einem Erkennungsvorgang, umfassend:

eine erste Signalquelle zum Bereitstellen eines ersten Bildes;

eine Bildeingabeschaltung zum Empfangen des Bildes von der ersten Bildquelle;

eine zweite Signalquelle zum Bereitstellen von Nichtbilddaten;

eine Eingabeschaltung zum Empfangen der Nichtbilddaten von der zweiten Signalquelle;

5 einen Speicher zum Speichern von Daten, die Instruktionen enthalten, die ein Prozessor ausführen kann; und

den Prozessor zum Empfangen des Bildes von der Bildeingabeschaltung und zum Empfangen der Nichtbilddaten von der Eingabeschaltung und zum Zugreifen auf die in dem Speicher gespeicherten Daten; wobei der Prozessor zum Durchführen der folgenden Schritte bei der Ausführung der Instruktionen eingerichtet ist:

10 Empfangen eines zweidimensionalen Quellenbildes (10, 50), das eine Vielzahl von Glyphproben enthält, von der Bildeingabeschaltung; wobei die Bildquelle eine vertikale Dimensionsgröße hat, die größer als eine einzelne Zeile von Glyphproben ist; wobei jede Glyphprobe ein Bildbeispiel eines jeweiligen Zeichens einer Vielzahl von Zeichen in dem Zeichensatz ist; wobei der Satz von Zeichenschablonen jeweilige Zeichen der Vielzahl von Zeichen in dem Zeichensatz repräsentiert;

Empfangen einer Transkriptionsdatenstruktur (40, 60, 70), die dem Quellenbild zugeordnet ist und eine geordnete Anordnung von Transkriptionslabeln (42, 44, 46, 47, 48, 62, 64, 66, 68, 72) enthält, von der Eingabeschaltung;

20 Bestimmen einer Glyphpixelposition für jede Glyphprobe;

Erzeugen eines mit jeder jeweiligen Glyphpixelposition gepaarten Glyphlabels; und

25 Erzeugen des Satzes von Zeichenschablonen unter Verwendung des Quellenbildes und der Glyphpixelpositionen und der jeweils gepaarten Glyphlabel; wobei jede Zeichenschablone durch eine Zeichenlabel identifiziert ist; wobei jedes Paar von Glyphpixelpositionen und Glyphlabeln eine Trainingsdatenprobe für eine jeweilige Zeichenschablone identifiziert, wenn das jeweils gepaarte Glyphlabel zu dem jeweiligen Zeichenlabel passt,

30 **dadurch gekennzeichnet, dass**

der Speicher weiterhin ein zweidimensionales Bildquellenmodell (820, 830, 770) zum Modellieren einer räumlichen Bildstruktur eines Satzes zweidimensionaler Bilder als eine Grammatik speichert; wobei das Bildquellenmodell räumliche Positionierungsdaten für eine Vielzahl von Glyphs enthält, die in einem ersten Bild auftreten, das in dem Satz von Bildern enthalten ist; wobei das Bildquellenmodell Mappingdaten zum Abbilden einer jeweiligen Glyphprobe, die in dem ersten Bild auftritt, auf ein Glyphlabel angibt, das ein Zeichen in einem Zeichensatz angibt; wobei das Quellenbild in dem Satz von Bildern enthalten ist; und

40 der Prozessor weiterhin eingerichtet ist, um die räumlichen Positionierungsdaten zur Bestimmung der Glyphpixelpositionen zu verwenden und um die Mappingdaten und die geordnete Anordnung von Transkriptionslabeln zur Erzeugung der gepaarten Glyphlabel zu verwenden.

9. Maschine nach Anspruch 8, weiterhin **dadurch gekennzeichnet, dass**

45 die erste Signalquelle eine Abtasteinrichtung ist; wobei die Bildeingabeschaltung eine Abtastschaltung ist, die ein Bild von einem Abtastvorgang empfangen kann, der auf einem physikalischen Dokument durchgeführt wird, das Markierungen zeigt, die das Bild angeben; wobei der Prozessor eingerichtet ist zum Empfangen des Quellenbildes von der Abtastschaltung; und/oder dass

50 die zweite Signalquelle eine Benutzereingabeeinrichtung zum Empfangen von Signalen von einem Benutzer enthält; wobei die Eingabeschaltung zum Empfangen der Signale von der Benutzereingabeeinrichtung verbunden ist; und

55 wobei der Prozessor zum Empfangen der Transkriptionsdatenstruktur von der Eingabeschaltung eingerichtet ist; und/oder dass

das Bildquellenmodell ein stochastisches Finitzustandsnetzwerk repräsentiert ist, das eine reguläre Gramma-

5 tik angibt und die Mappingdaten, die eine jeweilige Glyphpixelposition in dem Quellenbild auf eine Glyphlabel abbilden, als einen Pfad durch das Bildquellennetzwerk repräsentiert; wobei der Pfad ihm zugeordnete Pfaddatenelemente angibt und vom Prozessor zugreifbar ist; wobei die Pfaddatenelemente Positionen und jeweils gepaarte Glyphlabel jeweiliger in dem Quellenbild enthaltener Glyphs angeben; und

die Transkriptionsdatenstruktur als ein Finitzustandsnetzwerk (800, 850, 852) repräsentiert wird, das eine Transkription als einen Transkriptionspfad durch das Transkriptionsnetzwerk modelliert, der eine geordnete Anordnung von Transkriptionslabeln angibt; und

10 der Prozessor eingerichtet ist, um bei der Bestimmung der Glyphpixelpositionen und der Erzeugung jeweils gepaarter Glyphlabel die folgenden Schritte durchzuführen:

15 Verschmelzen des Bildquellennetzwerks mit dem Transkriptionsnetzwerk, um ein Transkriptionsbildnetzwerk (870, 872) zu erzeugen, das modifizierte Mappingdaten zum Abbilden eines jeweiligen Transkriptionslabels auf eine Glyphpixelposition eines jeweiligen in dem Quellenbild auftretenden Glyphs angibt; wobei die modifizierten Mappingdaten als wenigstens ein vollständiger Transkriptionsbildpfad durch das Transkriptionsbildnetzwerk repräsentiert ist; wobei der Transkriptionsbildpfad die Pfaddatenelemente angibt;

20 Dekodieren des Quellenbildes unter Verwendung des Transkriptionsbildnetzwerks, um den Transkriptionsbildpfad zu erzeugen; und

Erlangen der Glyphpixelposition jeder Glyphprobe und des damit gepaarten Glyphlabels unter Verwendung der Pfaddatenelemente, die durch den Transkriptionsbildpfad angegeben sind.

25 10. Maschine nach Anspruch 8, weiterhin **dadurch gekennzeichnet**, dass

30 die Transkriptionsdatenstruktur eine Tag-Transkriptionsdatenstruktur ist, die wenigstens ein nichtwörtliches Tag-Transkriptionslabel enthält, das wenigstens einen Zeichenkode angibt, der ein Zeichen repräsentiert, mit dem keine Glyphprobe durch visuelle Betrachtung derselben gepaart werden kann; wobei der durch das Tag-Transkriptionslabel angegebene Zeichenkode Markup-Informationen über das Quellenbild angibt; wobei die Markup-Informationen bei ihrer Interpretation durch einen Dokumentenverarbeitungsvorgang wenigstens ein in dem Quellenbild enthaltenes Displaymerkmal erzeugen, das als eine visuelle Formatierungscharakteristik wahrnehmbar ist; und

35 der Prozessor eingerichtet ist, um bei der Erzeugung der gepaarten Glyphlabel die räumlichen Positionierungsdaten zu verwenden, um wenigstens eine Glyphprobe zu identifizieren, die mit dem Tag-Transkriptionslabel in Beziehung steht, und um das Tag-Transkriptionslabel zu verwenden.

40 **Revendications**

1. Procédé destiné à l'apprentissage d'un ensemble (20) de gabarits de caractères (22, 24, 26) destiné à être utilisé dans un système de reconnaissance, chaque gabarit de caractère représentant un caractère respectif parmi une pluralité de caractères dans un ensemble de caractères et étant identifié par une étiquette de caractère indiquant le caractère respectif, le procédé étant destiné à mettre en oeuvre une machine comprenant un processeur et un dispositif de mémoire relié à celui-ci, le dispositif de mémoire mémorisant des données d'instructions que le processeur exécute, le procédé comprenant les étapes consistant à :

50 recevoir une image source bidimensionnelle (10, 50) comprenant une pluralité d'échantillons de glyphes (12, 14), chaque échantillon de glyphe étant une instance d'une image d'un caractère respectif parmi la pluralité de caractères dans l'ensemble de caractères, l'image source présentant une taille de dimension verticale plus grande que la hauteur d'une seule ligne d'échantillons de glyphes,

recevoir une structure de données de transcription (40, 60, 70) qui est associée à l'image source et comprenant un agencement ordonné d'étiquettes de transcription (42, 44, 46, 47, 48, 62, 64, 66, 68, 72),

55 déterminer (220) une position de pixel de glyphe de chaque échantillon de glyphe,

produire (250) pour chaque échantillon de glyphe une étiquette de glyphe indiquant un caractère respectif parmi les caractères de l'ensemble de caractères et apparier (250) celle-ci à la position de pixel de glyphe respective, et

produire (270, 400) l'ensemble des gabarits de caractères en utilisant les paires de positions de pixels de glyphes et les étiquettes de glyphes et en utilisant les échantillons de glyphes aux positions de pixels de glyphes en tant qu'échantillons de données d'apprentissage pour un gabarit respectif parmi les gabarits de caractères,

caractérisé en ce que

le dispositif de mémoire mémorise en outre un modèle de source d'image bidimensionnelle (820, 830, 770) destiné à modéliser en tant que grammaire une structure d'image dans l'espace d'un ensemble de deux images bidimensionnelles, le modèle de source d'image comprenant des données de positionnement dans l'espace destinées à modéliser le positionnement dans l'espace d'une pluralité de glyphes incluse dans une image et mapper des données en vue du mappage d'un glyphe respectif parmi les glyphes sur une étiquette de glyphe indiquant le caractère dans l'ensemble de caractères, l'image source étant une image de l'ensemble d'images, l'étape de détermination des positions de pixels de glyphes utilise les données de positionnement dans l'espace, et l'étape de production et d'appariement des étiquettes de glyphes utilise les données de mappage et la structure de données de transcription.

2. Procédé selon la revendication 1, caractérisé en outre en ce que chaque position de pixel de glyphe est une position d'origine unique de l'échantillon de glyphe respectif et l'étape de production (270, 400) de l'ensemble de gabarits de caractères comprend les étapes consistant à :

déterminer (430) des régions dans l'image source, chaque région comprenant une pluralité de positions de pixels d'échantillons et étant suffisamment grande pour que deux des positions de pixels d'échantillons constituent les positions d'origines de deux échantillons de glyphes, et affecter (450) des valeurs de couleurs de pixels à des positions de pixels de gabarits incluses dans des gabarits respectifs parmi les gabarits de caractères en utilisant des valeurs de couleurs de pixels indiquées par les positions de pixels d'échantillons sur la base de critères qui déterminent quelles positions de pixels d'échantillons dans lesquelles des régions sont utilisées pour déterminer une valeur de couleur de pixel pour une position de pixel de gabarit de sorte que tous les gabarits de caractères vérifient un modèle de gabarit de caractère conformément auquel deux gabarits de caractères adjacents qui sont décalés l'un par rapport à l'autre d'une largeur de positionnement de caractères peuvent comporter des cadres de limitation en chevauchement mais pratiquement aucun pixel d'avant-plan en chevauchement, chaque cadre de limitation contenant entièrement un gabarit de caractère.

3. Procédé selon la revendication 1 ou la revendication 2, caractérisé en outre en ce que la structure de données de transcription est une structure de données de transcription de repère comprenant au moins une étiquette de transcription de repère non littérale indiquant au moins un code de caractère représentant un caractère avec lequel aucun échantillon de glyphe dans l'image source ne peut être apparié par un examen visuel de celle-ci, le au moins un code de caractère indiquant des informations de marquage concernant l'image source, les informations de marquage, lorsqu'elles sont interprétées par une opération de traitement de document, étant capables de produire au moins une caractéristique d'affichage incluse dans l'image source perceptible sous forme d'une caractéristique de formatage visuelle, et dans lequel l'étape de production (250) de l'étiquette de glyphe utilise les données de positionnement dans l'espace concernant la pluralité des échantillons de glyphes pour identifier au moins un échantillon de glyphe associé à la structure de données de repères, et dans lequel l'étape d'appariement (250) utilise la structure de données de repères pour l'appariement.

4. Procédé destiné à l'apprentissage d'un ensemble de gabarits de caractères en vue d'une utilisation dans un système de reconnaissance, chaque gabarit de caractère représentant un caractère respectif parmi une pluralité de caractères dans un ensemble de caractères, le procédé étant destiné à mettre en oeuvre une machine comprenant un processeur et un dispositif de mémoire qui lui est relié, le dispositif de mémoire mémorisant des données d'instructions qu'exécute le processeur, le procédé comprenant les étapes consistant à :

recevoir et mémoriser (290) une image source bidimensionnelle comprenant une pluralité d'échantillons de glyphes, chaque échantillon de glyphe étant une instance d'une image d'un caractère respectif de la pluralité de caractères dans l'ensemble de caractères, l'image source présentant une taille de dimension verticale plus grande que la hauteur d'une seule ligne d'échantillons de glyphes, recevoir et mémoriser (292) un réseau de transcription à états finis (800, 850, 852) qui est associé à l'image

source et comprenant un agencement ordonné d'étiquettes de transcription sous forme d'au moins un trajet de transcription au travers du réseau de transcription, décoder (330) l'image source, obtenir (380) pour chaque échantillon de glyphe dans l'image source, une position et une étiquette de glyphe qui lui est appariée, et produire (400) l'ensemble des gabarits de caractères en utilisant l'image source et les paires de positions et d'étiquettes de glyphes et en utilisant les échantillons de glyphes identifiés par les étiquettes de glyphes pour l'apprentissage des gabarits de caractères,

caractérisé en ce que

la mémoire mémorise en outre un réseau de source d'image à états finis stochastique bidimensionnelle (830) destiné à modéliser en tant que grammaire une structure d'image dans l'espace d'un ensemble d'images bidimensionnelles, chacune comprenant une pluralité de glyphes, une première image de l'ensemble d'images étant modélisée sous forme d'au moins un trajet au travers du réseau de source d'image qui indique une image idéale cohérente avec la structure d'image dans l'espace de la première image, le au moins un trajet indiquant des éléments de données de trajet qui lui sont associés, les éléments de données de trajet indiquant des positions et des étiquettes de glyphes appariées avec celles-ci des glyphes respectifs parmi la pluralité de glyphes inclus dans la première image, l'image source étant une image de l'ensemble des images, le procédé comprenant en outre une étape consistant à fusionner (300) le réseau de source d'image avec le réseau de transcription afin de produire un réseau de transcription-image (870, 872) dans lequel, lorsque la transcription est associée à la première image, le réseau de transcription-image modélise la première image comme étant un trajet de transcription-image complet au travers du réseau de transcription-image qui indique une image idéale cohérente avec la structure d'image dans l'espace de la première image, le au moins un trajet de transcription-image complet indiquant les éléments de données de trajet et une séquence de chaînes de message cohérente avec l'agencement ordonné des étiquettes de transcription, l'étape de décodage utilise le réseau de transcription-image pour produire au moins un trajet de transcription-image complet indiquant une image idéale cohérente avec la structure d'image dans l'espace de l'image source, et l'étape d'obtention utilise les éléments de données de trajet associés à au moins un trajet de transcription-image complet.

5. Procédé selon la revendication 4, caractérisé en outre en ce que

chaque gabarit de caractère est fondé sur un modèle à portée latérale de positionnement d'image de caractère de deux images de caractères adjacents qui sont décalées l'une de l'autre d'une largeur de positionnement de caractère, le modèle à portée latérale nécessitant que, lorsque deux cadres de limitation rectangulaires contenant chacun l'une des deux images de caractères se chevauchent, les images de caractères ne comportent pratiquement aucun pixel d'avant-plan en chevauchement, et dans lequel l'étape de production (400) de l'ensemble de gabarits de caractères comprend l'étape consistant à déterminer (490) pour chaque gabarit de caractère, une largeur de positionnement de caractère de celui-ci en utilisant les paires de positions et d'étiquettes de glyphes, ou bien en ce que chacun des gabarits de caractères est fondé sur un modèle à base de segmentation de positionnement d'image de caractère de deux images de caractères adjacents, dont chacune est capable d'être contenue entièrement dans l'un de deux cadres de limitation rectangulaires pratiquement sans chevauchement.

6. Procédé selon la revendication 4, caractérisé en outre en ce que

le réseau de transcription comprend au moins une paire d'étiquettes de transcription indiquant au moins des première et seconde chaînes de message différentes pour une seule partie d'image de l'image source, le réseau de transcription comprenant une série de noeuds de transcription et une séquence de transitions entre des paires des noeuds, chaque transition comportant une étiquette de transcription qui lui est associée, le réseau de transcription modélisant la au moins une paire d'étiquettes de transcription sous forme d'au moins des première et seconde séquences différentes de transitions entre l'une des paires des noeuds, chacune des au moins première et seconde séquences différentes de transitions comportant l'une des au moins première et seconde chaînes de message différentes qui leur sont associées, ou en ce que le réseau de transcription indique une transcription de repère comprenant au moins une étiquette de transcription de repère non littérale indiquant au moins un code de caractère représentant un caractère avec lequel

un glyphe respectif dans l'image source ne peut pas être apparié par un examen visuel de celle-ci, le au moins un code de caractère indiquant des informations de marquage concernant l'image source, les informations de marquage, lorsqu'elles sont interprétées par une opération de traitement de document, étant capables de produire au moins une caractéristique d'affichage incluse dans l'image source perceptible sous forme d'une caractéristique de formatage visuelle, et l'étape de fusion (300) comprenant l'étape de désignation des étiquettes de transcription sous forme de chaînes de message incluses dans le réseau de transcription-image, l'étiquette de transcription de repère étant conçue sous forme d'une chaîne de message de sorte que le réseau de transcription-image modélise une relation entre l'étiquette de transcription de repère et au moins un glyphe, ou en ce que

le au moins un trajet de transcription-image complet est défini par une séquence de transitions entre des noeuds inclus dans le réseau de transcription-image, les éléments de données de trajet associés à celui-ci étant associés à chaque transition et comprenant une chaîne de message, un gabarit de caractère, et un décalage d'image, et l'étape d'obtention (380) comprend l'étape consistant à déterminer, pour les transitions respectives comportant des gabarits de caractères non nuls qui leur sont associés, les positions des glyphes respectifs en utilisant les décalages d'image associés aux transitions respectives et en outre de détermination de l'étiquette de glyphe appariée avec chaque position respective en utilisant une étiquette de caractère indiquée par le gabarit de caractère non nul associé à la transition respective, l'étiquette de caractère indiquant un caractère dans l'ensemble de caractères.

7. Procédé selon la revendication 4, caractérisé en outre en ce que l'étape de décodage (330) comprend les étapes consistant à :

exécuter (334, 338, 340) une opération de décodage fondée sur une programmation dynamique pour calculer une évaluation optimum à chaque noeud d'une pluralité de noeuds d'un treillis dans un treillis de décodage représentant le réseau de transcription-image, ladite opération de décodage produisant pour chaque noeud du treillis un élément de données d'identification de transition d'optimisation en tant que résultat du calcul de l'évaluation optimum, mémoriser l'élément et indiquer l'une d'une pluralité de transitions possibles dans un noeud respectif parmi les noeuds du treillis qui optimise l'évaluation pour le noeud de treillis respectif, et retracer vers l'arrière (380) pour récupérer une séquence de transitions indiquant un trajet de treillis de décodage, ladite opération de tracé vers l'arrière commençant par un noeud final du treillis et se terminant par un premier noeud du treillis dans le trajet de treillis de décodage, la séquence des transitions étant récupérée en utilisant l'élément de données, le trajet de treillis de décodage indiquant le au moins un trajet de transcription-image complet au travers du réseau de transcription-image.

8. Machine destinée à l'apprentissage d'un ensemble de gabarits de caractères en vue d'une utilisation dans une opération de reconnaissance, comprenant :

une première source de signaux destinée à fournir une première image,
des circuits d'entrée d'image destinés à recevoir ladite image de la première source de signaux,
une seconde source de signaux destinée à fournir des données qui n'appartiennent pas à l'image,
des circuits d'entrée destinés à recevoir lesdites données qui n'appartiennent pas à l'image de la seconde source de signaux,
une mémoire destinée à mémoriser des données comprenant des instructions qu'un processeur peut exécuter, et

le processeur qui est destiné à recevoir ladite image des circuits d'entrée d'image et à recevoir lesdites données n'appartenant pas à l'image des circuits d'entrée et ainsi qu'à accéder aux données mémorisées dans la mémoire, le processeur étant agencé pour, en exécutant les instructions, recevoir, à partir des circuits d'entrée d'image, une image source bidimensionnelle (10, 50) comprenant une pluralité d'échantillons de glyphes, l'image source présentant une taille de dimension verticale plus grande qu'une seule ligne d'échantillons de glyphes, chaque échantillon de glyphe étant une instance d'une image d'un caractère respectif d'une pluralité de caractères dans l'ensemble de caractères, l'ensemble de gabarits de caractères représentant des caractères respectifs de la pluralité de caractères dans l'ensemble de caractères, recevoir depuis les circuits d'entrée une structure de données de transcription (40, 60, 70) qui est associée à l'image source et comprenant un agencement ordonné d'étiquettes de transcription (42, 44, 46, 47, 48, 62, 64, 66, 68, 72), déterminer une position de pixel de glyphe pour chaque échantillon de glyphe, produire une étiquette de glyphe appariée avec chaque position de pixel de glyphe respective, et

produire l'ensemble des gabarits de caractères en utilisant l'image source et les positions de pixels de glyphes et les étiquettes de glyphes respectivement appariées, chaque gabarit de caractère étant identifié par une étiquette de caractère, chaque paire de positions de pixels de glyphes et d'étiquettes de glyphes identifiant un échantillon de données d'apprentissage pour un gabarit respectif parmi les gabarits de caractères lorsque l'étiquette de glyphe appariée respectivement correspond à l'étiquette de caractère respective,

caractérisée en ce que

la mémoire mémorise en outre un modèle de source d'image bidimensionnelle (820, 830, 770), destiné à modéliser en tant que grammaire une structure d'image dans l'espace d'un ensemble d'images bidimensionnelles, le modèle de source d'image comprenant des données de positionnement dans l'espace pour une pluralité de glyphes apparaissant dans une première image incluse dans l'ensemble des images, le modèle de source d'image indiquant des données de mappage en vue de mapper un échantillon de glyphe respectif apparaissant dans la première image sur une étiquette de glyphe indiquant un caractère dans un ensemble de caractères, l'image source étant incluse dans l'ensemble des images, et le processeur est agencé en outre pour utiliser les données de positionnement dans l'espace en vue de déterminer les positions de pixels de glyphes et d'utiliser les données de mappage et ainsi que l'agencement ordonné des étiquettes de transcription en vue de produire les étiquettes de glyphes appariées.

9. Machine selon la revendication 8, caractérisée en outre en ce que

la première source de signaux est un dispositif de numérisation, les circuits d'entrée d'image sont des circuits de numérisation capables de recevoir une image grâce à une opération de numérisation exécutée sur un document physique présentant des repères indiquant l'image, le processeur est agencé pour recevoir l'image source depuis les circuits de numérisation, et/ou en ce que

la seconde source de signaux comprend un dispositif de saisie de l'utilisateur destiné à recevoir des signaux provenant d'un utilisateur, les circuits d'entrée étant reliés de façon à recevoir les signaux provenant du dispositif de saisie de l'utilisateur, et le processeur est agencé pour recevoir la structure de données de transcription depuis les circuits d'entrée, et/ou en ce que

le modèle de source d'image s'est représenté sous forme d'un réseau à états finis stochastique indiquant une grammaire de règles et représentant les données de mappage qui mappent une position de pixel de glyphe respective dans l'image source sur une étiquette de glyphe sous forme d'un trajet au travers du réseau de sources d'images, le trajet indiquant des éléments de données de trajet associés à celui-ci, et qui sont accessibles au processeur, les éléments de données de trajet indiquant des positions et des étiquettes de glyphes appariées respectivement des glyphes respectifs inclus dans l'image source, et

la structure de données de transcription est représentée sous forme d'un réseau à états finis (800, 850; 852) modélisant une transcription en tant que trajet de transcription au travers du réseau de transcription indiquant l'agencement ordonné des étiquettes de transcription, et

le processeur est agencé pour, lors de la détermination des positions de pixels de glyphes et la production des étiquettes de glyphes appariées respectivement,

fusionner le réseau de source d'image avec le réseau de transcription afin de produire un réseau de transcription-image (870, 872) indiquant des données de mappage modifiées en vue du mappage d'une étiquette de transcription respective sur une position de pixel de glyphe d'un glyphe respectif apparaissant dans l'image source, les données de mappage modifiées étant représentées sous forme d'au moins un trajet de transcription-image complet au travers du réseau de transcription-image, le trajet de transcription-image indiquant les éléments de données de trajet,

décoder l'image source en utilisant le réseau de transcription-image afin de produire le trajet de transcription-image, et

obtenir la position de pixel de glyphe de chaque échantillon de glyphe et l'étiquette de glyphe appariée avec celui-ci en utilisant les éléments de données de trajet indiqués par le trajet de transcription-image.

10. Machine selon la revendication 8, caractérisée en outre en ce que

la structure de données de transcription est une structure de données de transcription à repères comprenant au moins une étiquette de transcription de repère non littérale indiquant au moins un code de caractère représentant un caractère avec lequel aucun échantillon de glyphe ne peut être apparié par un examen visuel de celui-ci, le code de caractère indiqué par l'étiquette de transcription de repère indiquant des informations de marquage concernant l'image source, les informations de marquage, lorsqu'elles sont interprétées par une

EP 0 740 263 B1

opération de traitement de document, produisant au moins une caractéristique d'affichage incluse dans l'image source, perceptible sous forme d'une caractéristique de formatage visuelle, et le processeur est agencé pour, lors de la production de l'étiquette de glyphe appariée, utiliser les données de positionnement dans l'espace pour identifier au moins un échantillon de glyphe associé à l'étiquette de transcription de repère et utiliser l'étiquette de transcription de repère.

5

10

15

20

25

30

35

40

45

50

55

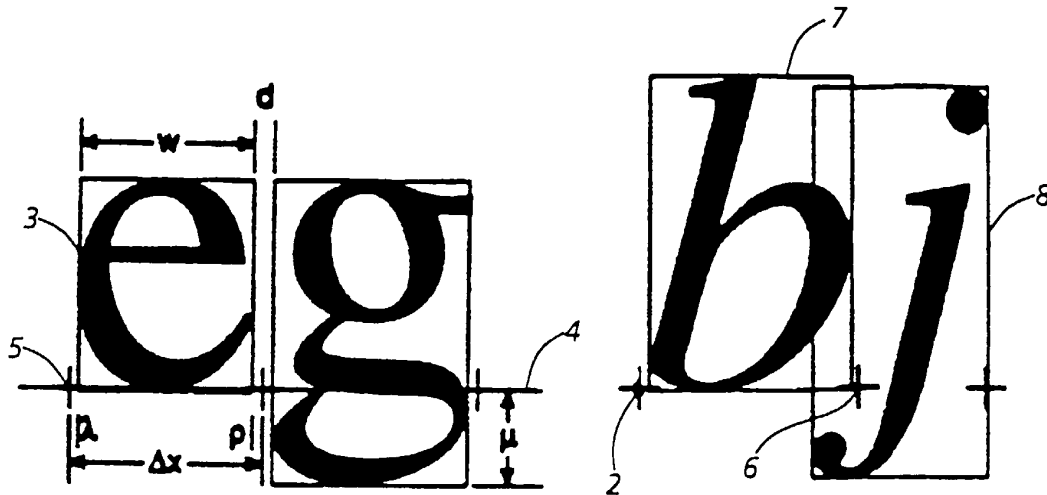


Fig. 1

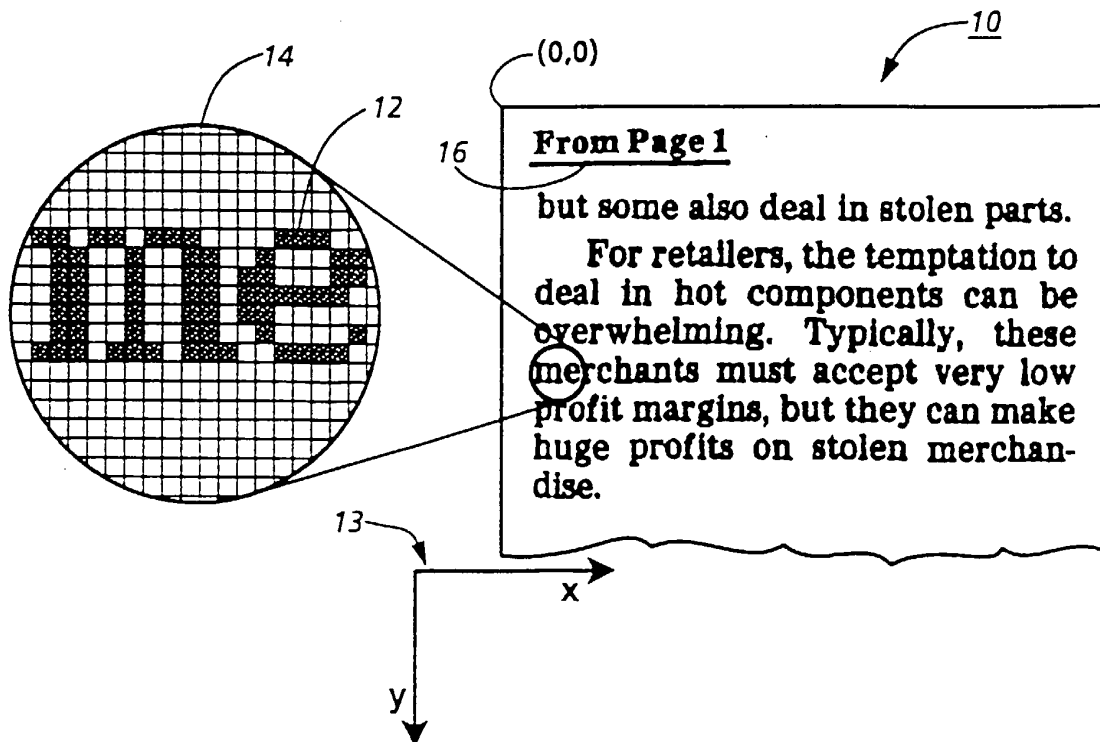


Fig. 2

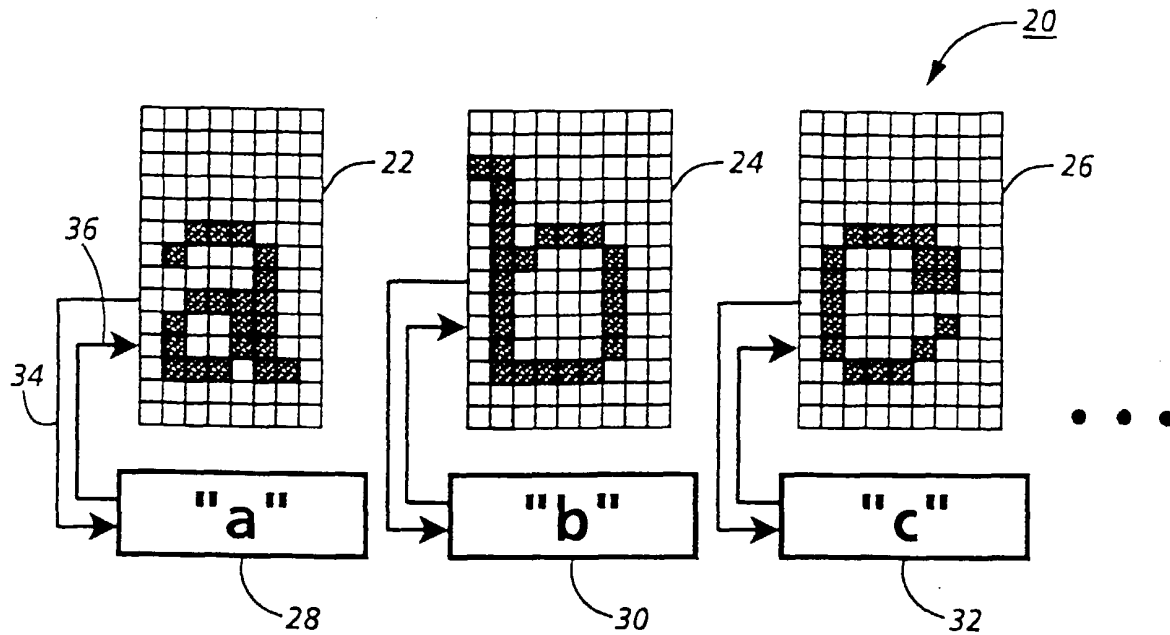


Fig. 3

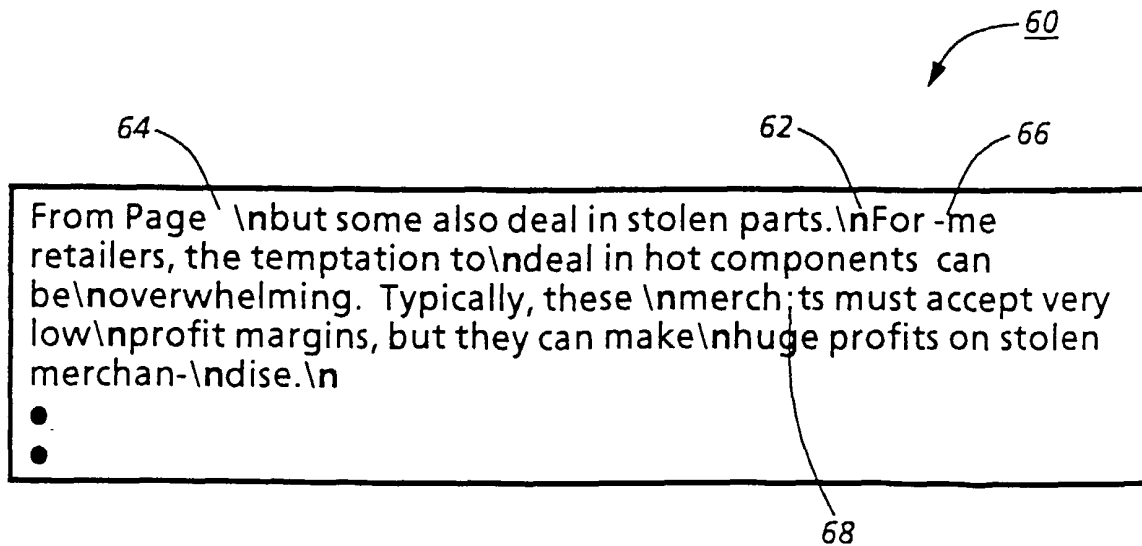


Fig. 4

40

```

47 \entry{
  \name{interference spectrum }
  \category{electr}
  \body{ Frequency distribution of the
42 jamming interference in the propagation medium external to the receiver
  }}
  \entry{
44 \name{interflow }
46 \category{hyd}
48 \body{ The water, derived from precipitation,
  that infiltrates the soil surface and then moves laterally through
  the upper layers of soil above the water table until it reaches a
  stream channel or returns to the surface at some point downslope from
  its point of infiltration
  }}
  \entry{
    \name{interlude }
    \category{adp}
    \body{ A small routine or program which is designed
    to carry out minor preliminary calculations or housekeeping operations
    before the main routine begins to operate and which can usually be
    overwritten after it has performed its function.
    }}
  
```

Fig. 5

50

54 55

52 **Interference spectrum** [*electr*] Frequency distribution of the jamming interference in the propagation medium external to the receiver.

Interflow [*hyd*] The water, derived from precipitation, that infiltrates the soil surface and then moves laterally through the upper layers of soil above the water table until it reaches a stream channel or returns to the surface at some point downslope from its point of infiltration.

Interlude [*adp*] A small routine or program which is designed to carry out minor preliminary calculations or housekeeping operations before the main routine begins to operate and which can usually be overwritten after it has performed its function.

Fig. 6

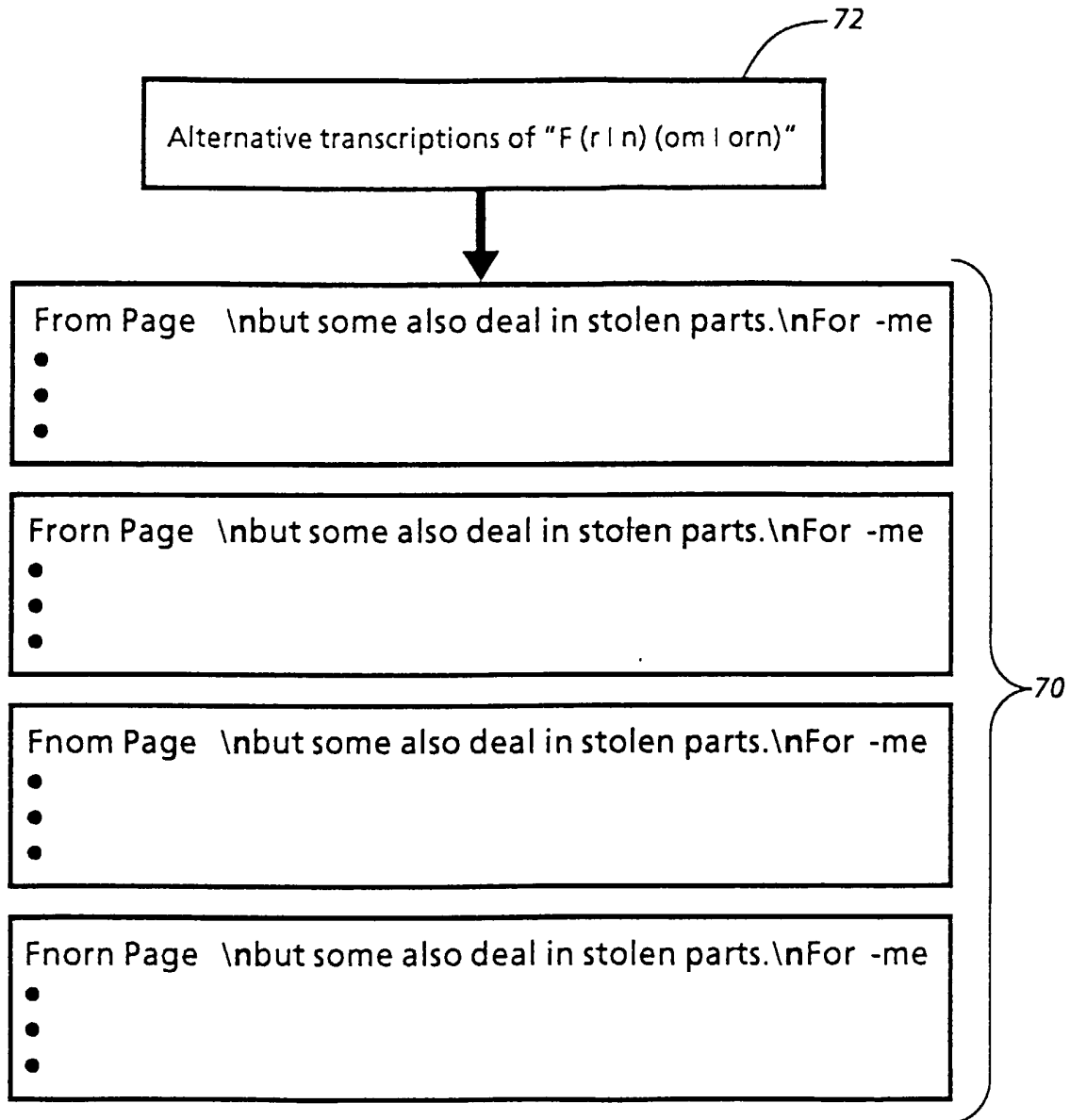


Fig. 7

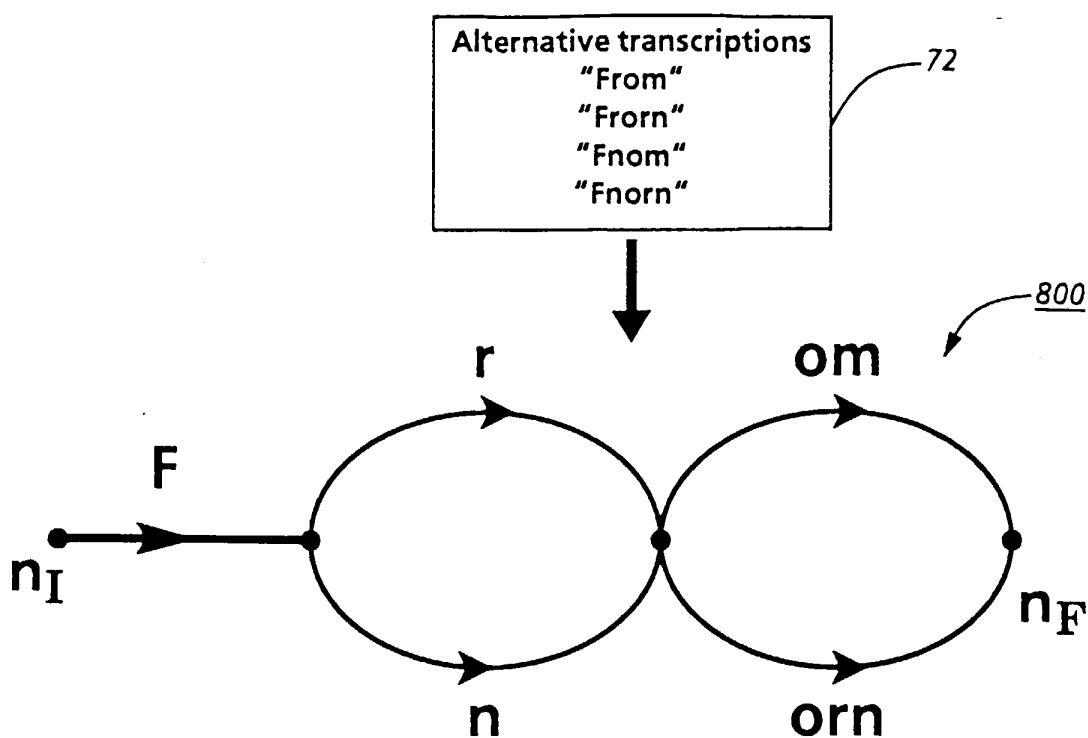


Fig. 8

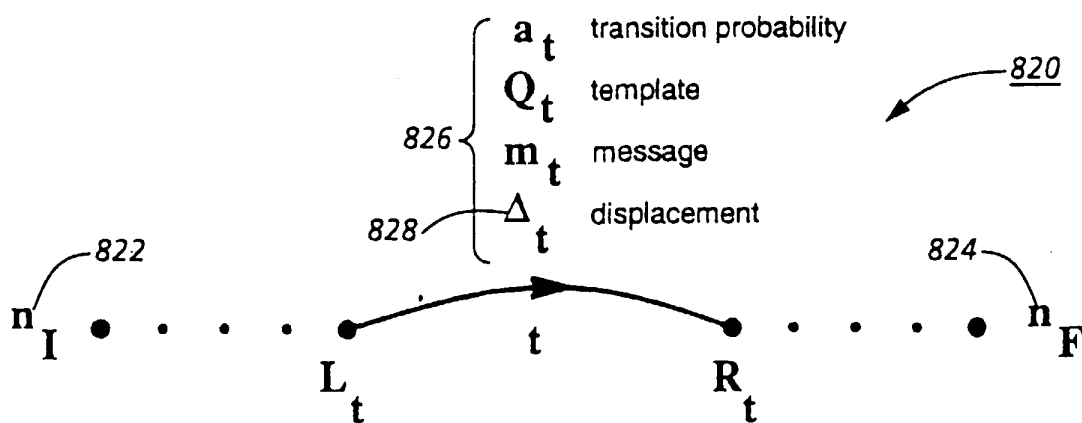
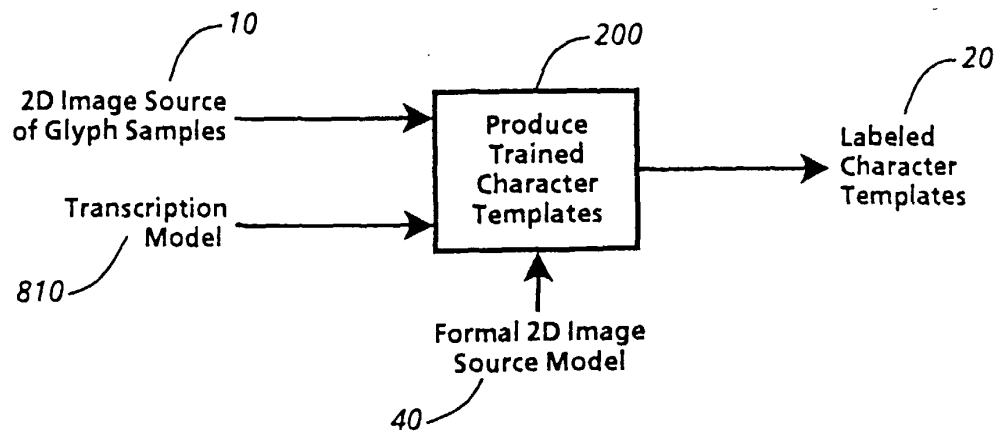
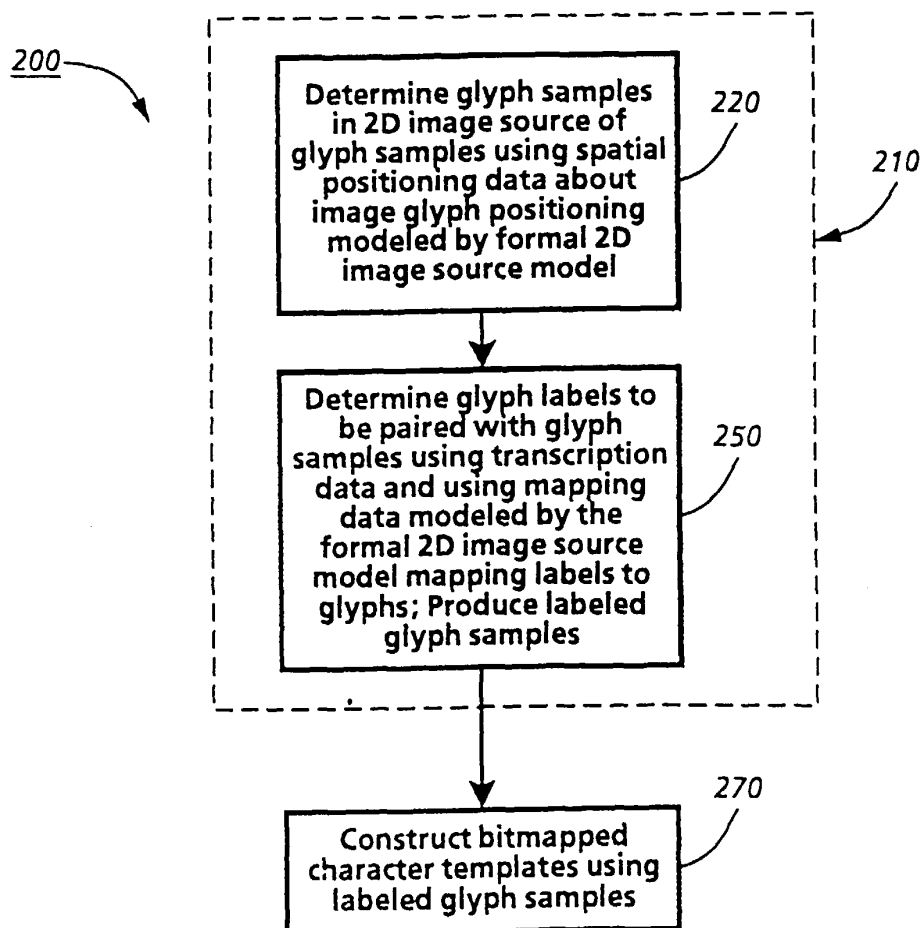
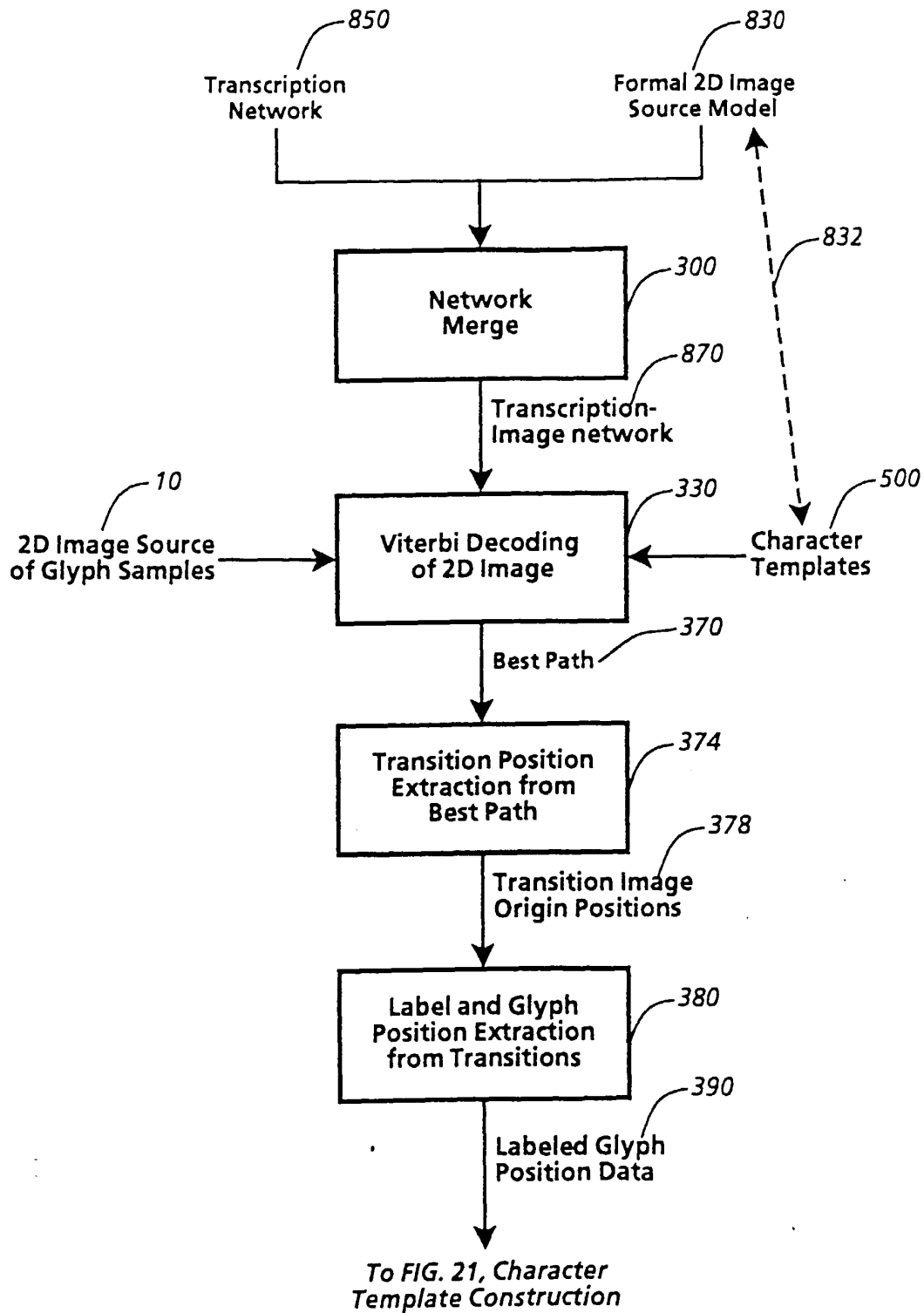
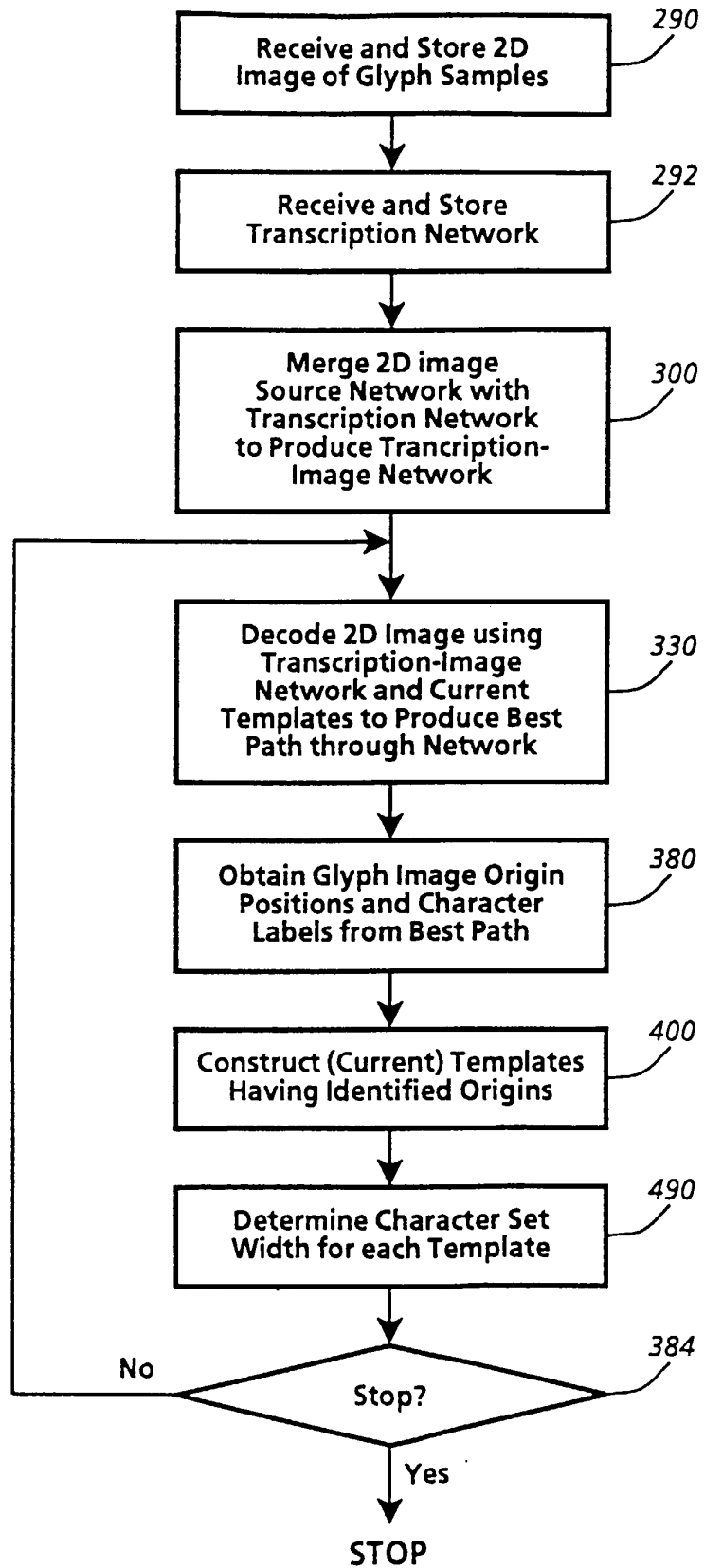


Fig. 9

**Fig. 10****Fig. 11**

**Fig.12**

**Fig.13**

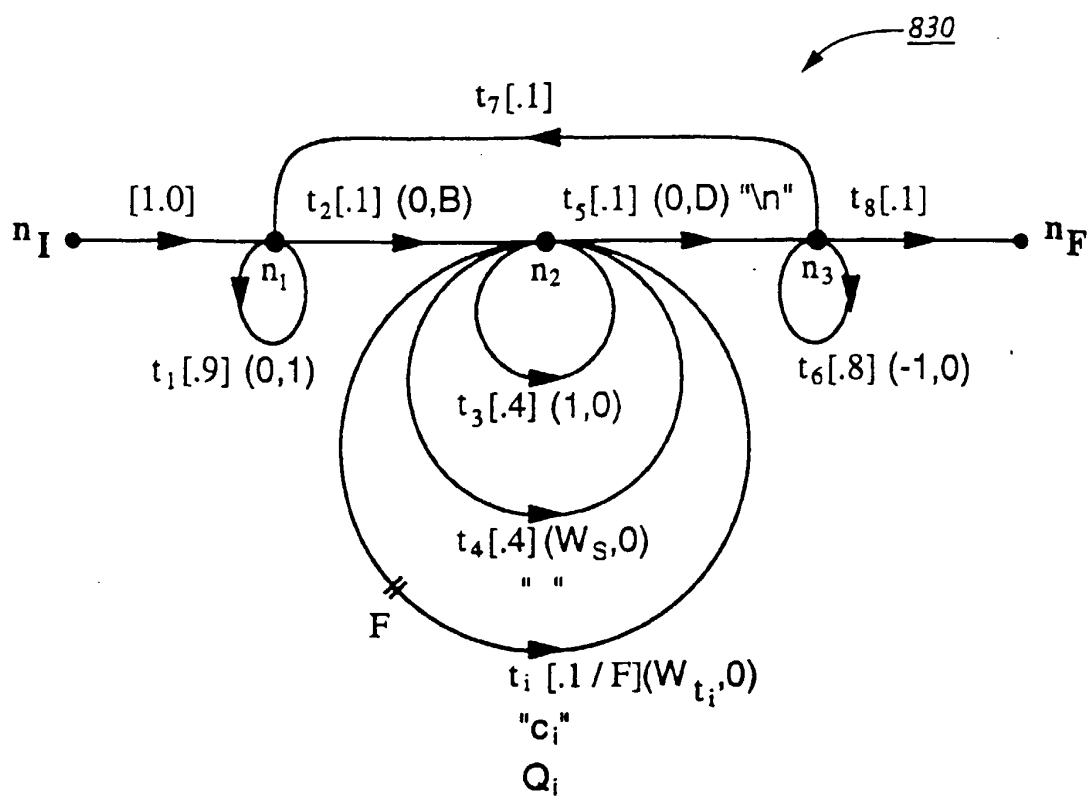


Fig.14

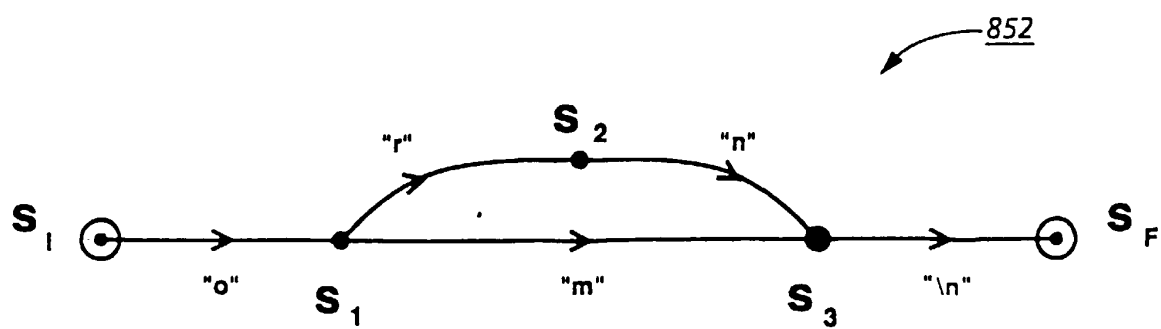


Fig.15

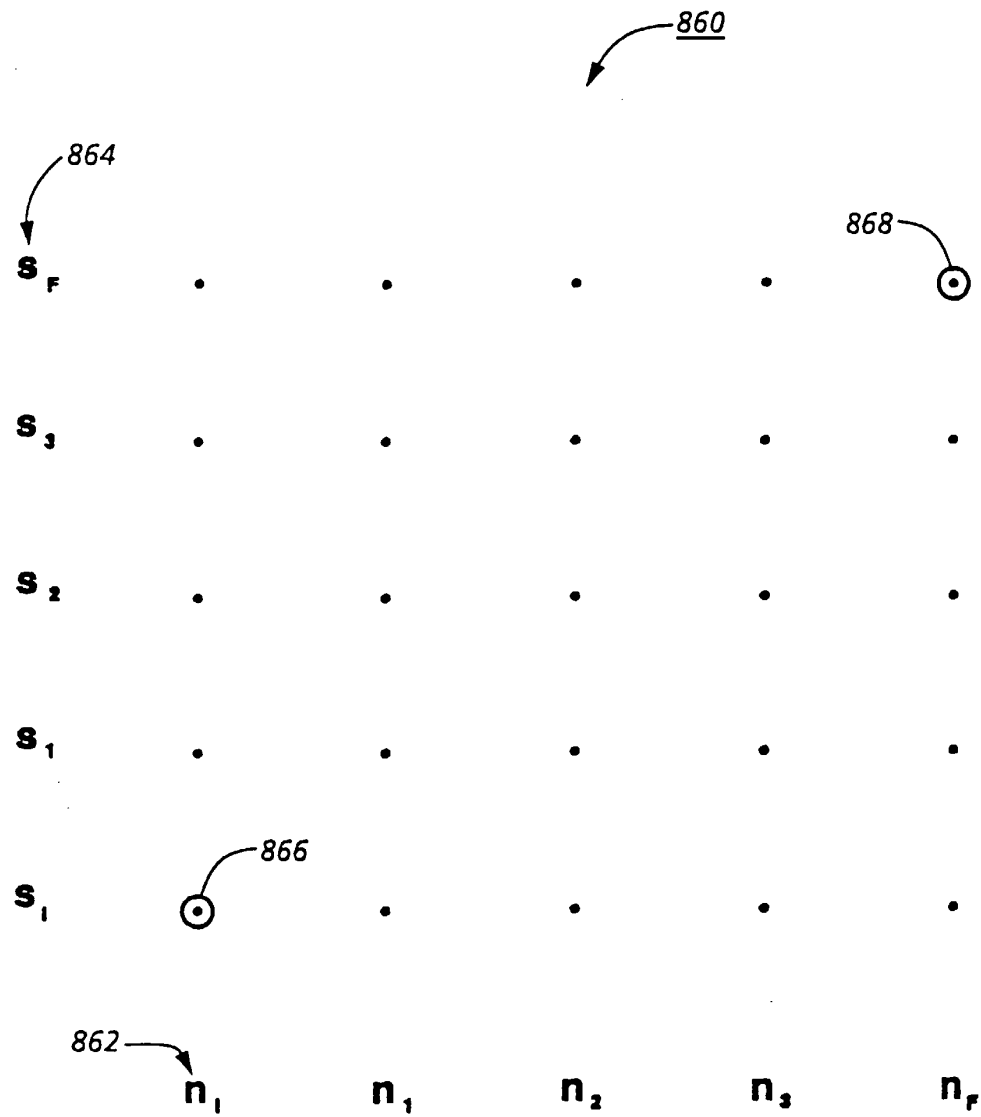


Fig.16

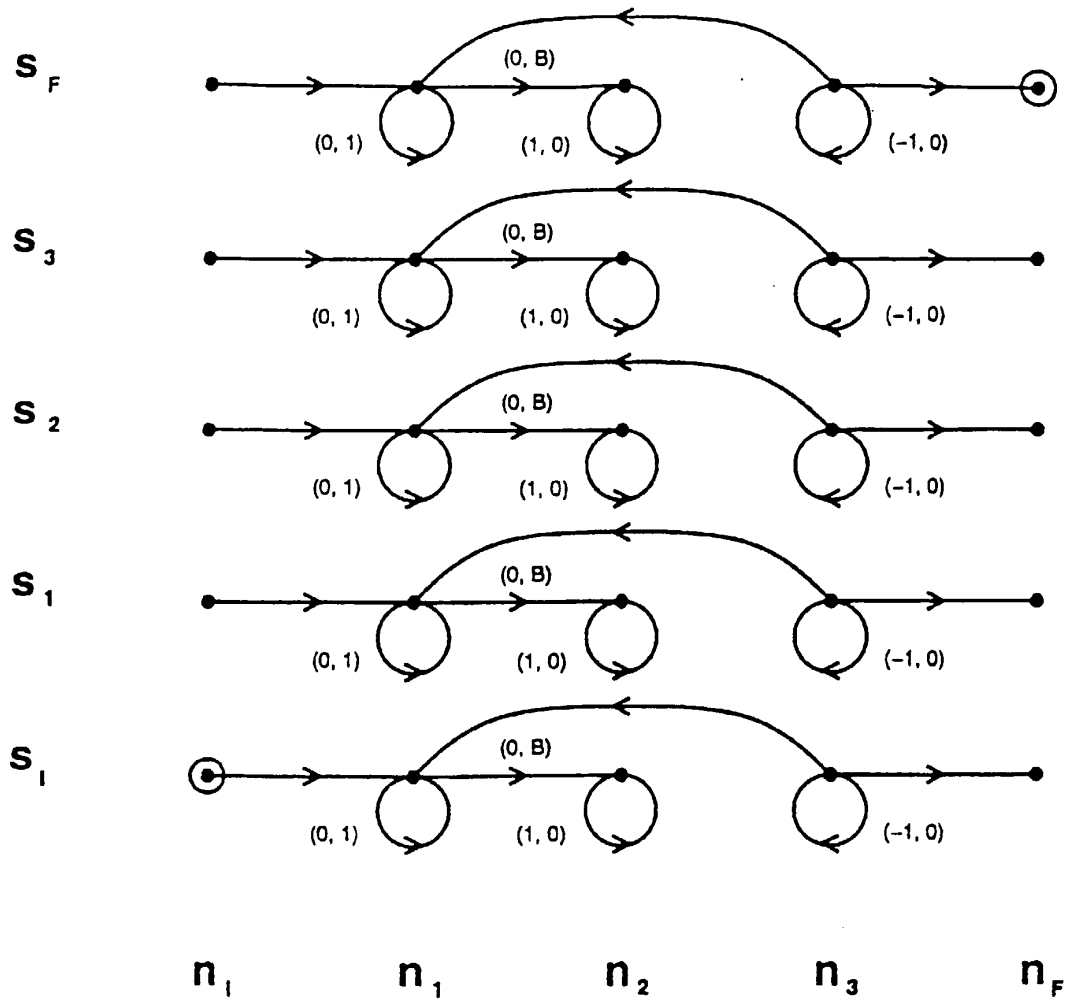


Fig.17

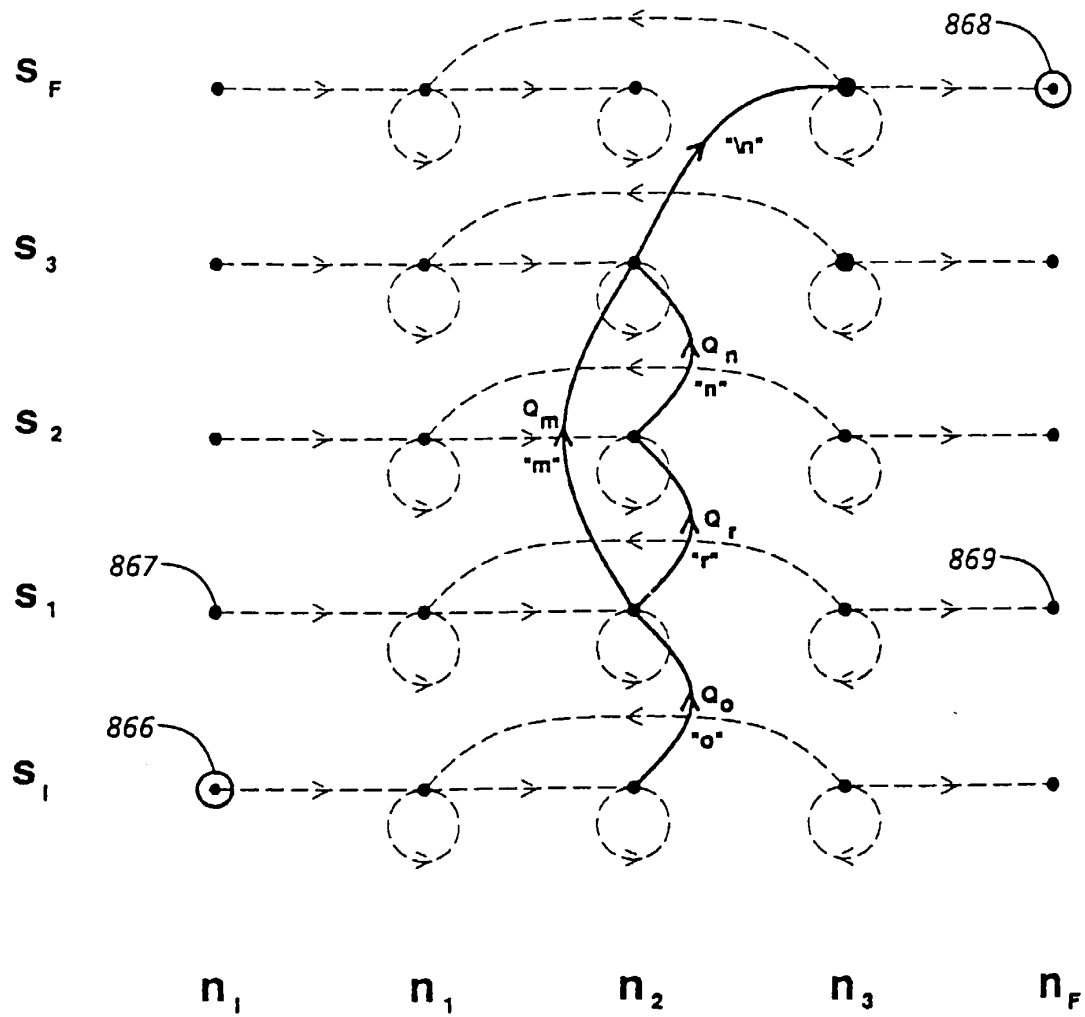


Fig.18

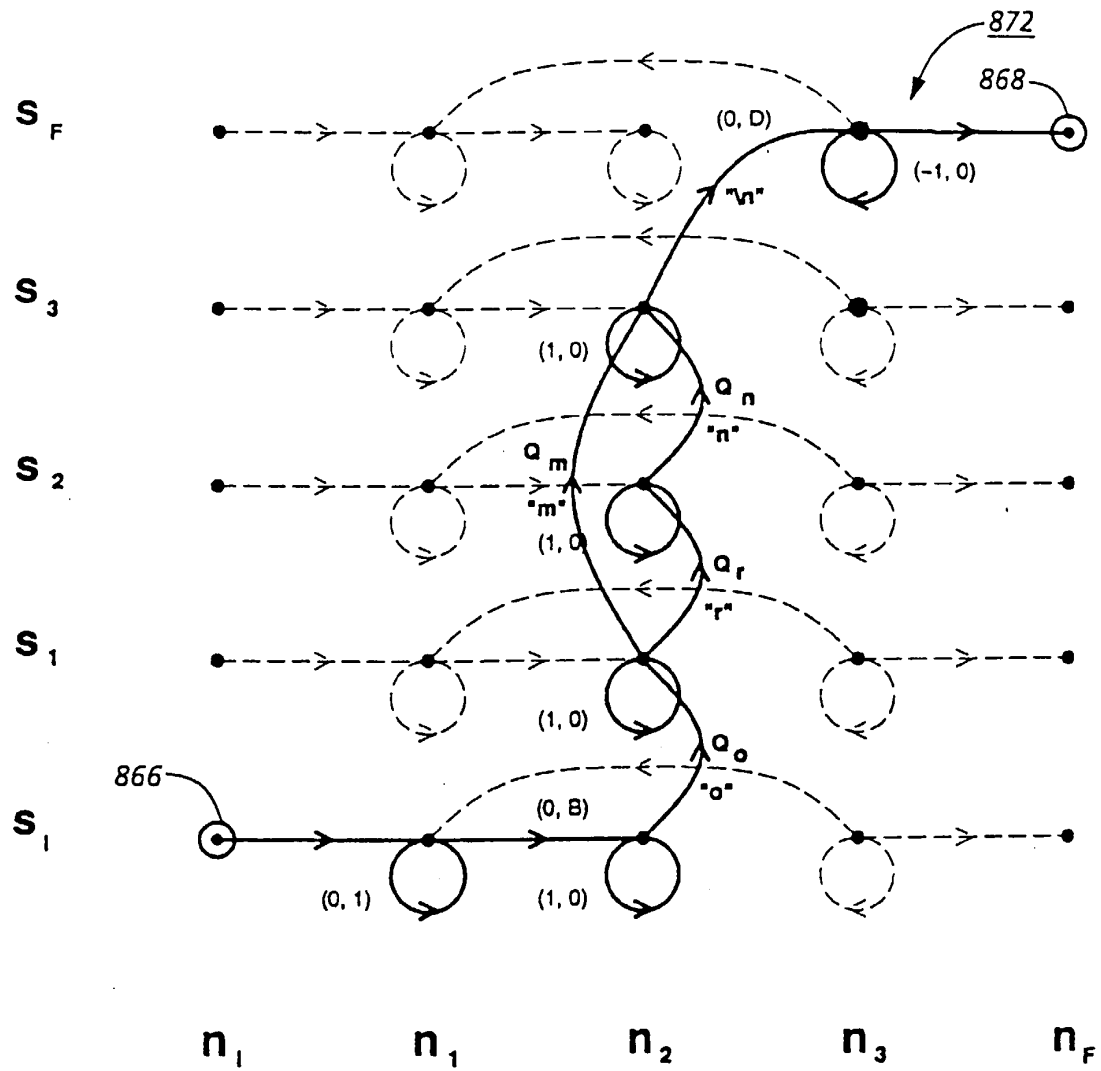
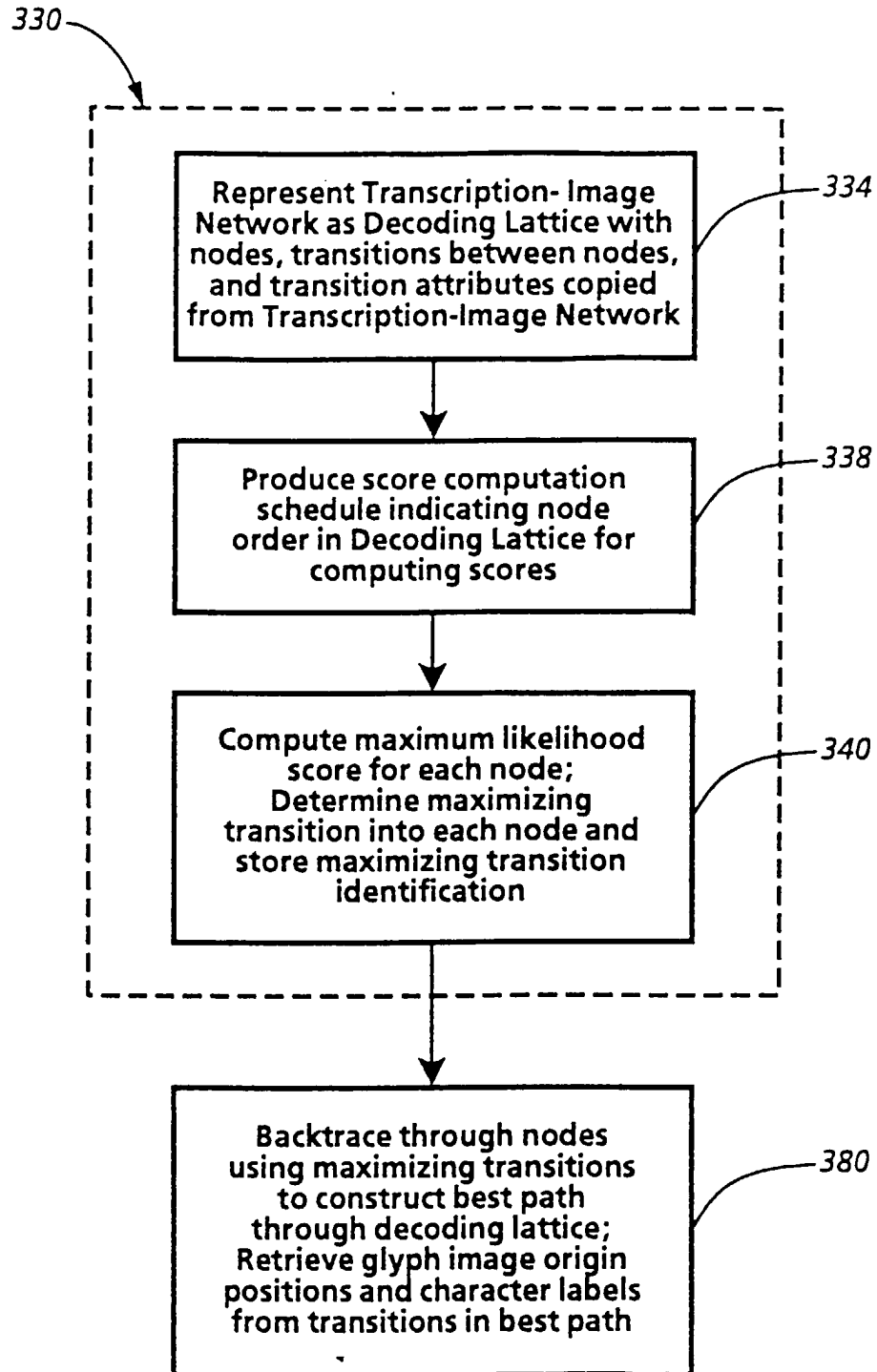
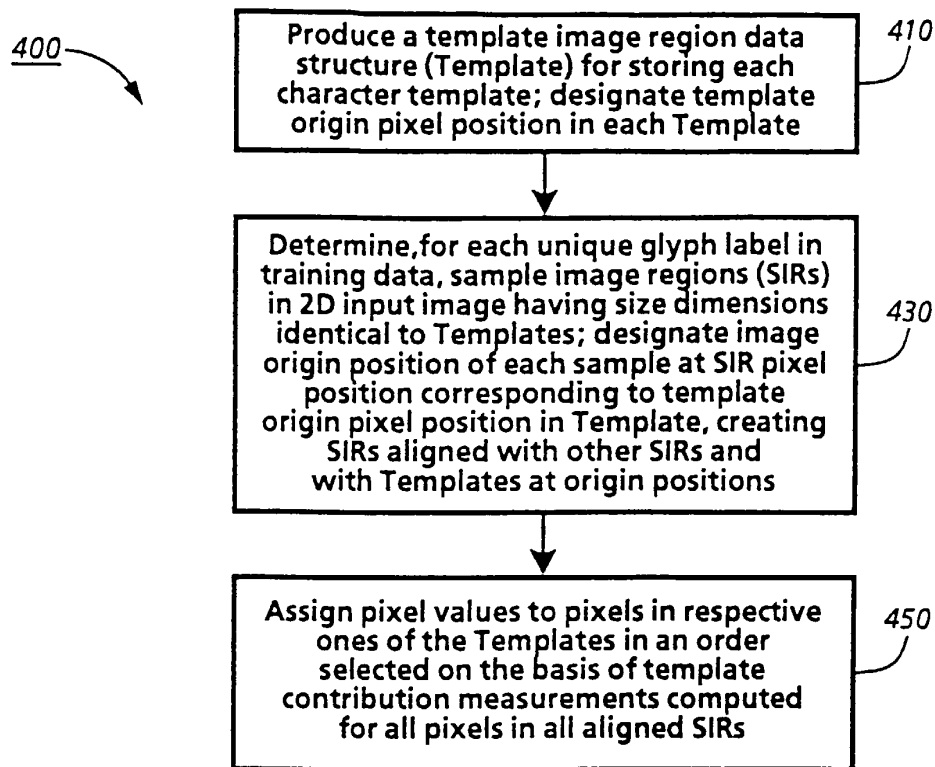
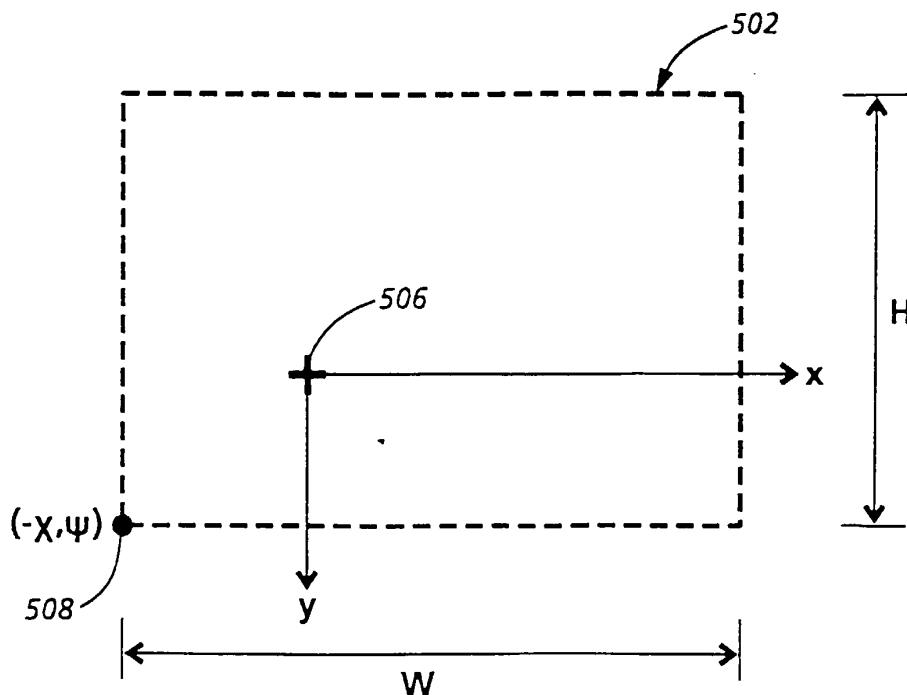


Fig.19

**Fig.20**

**Fig.21****Fig.22**

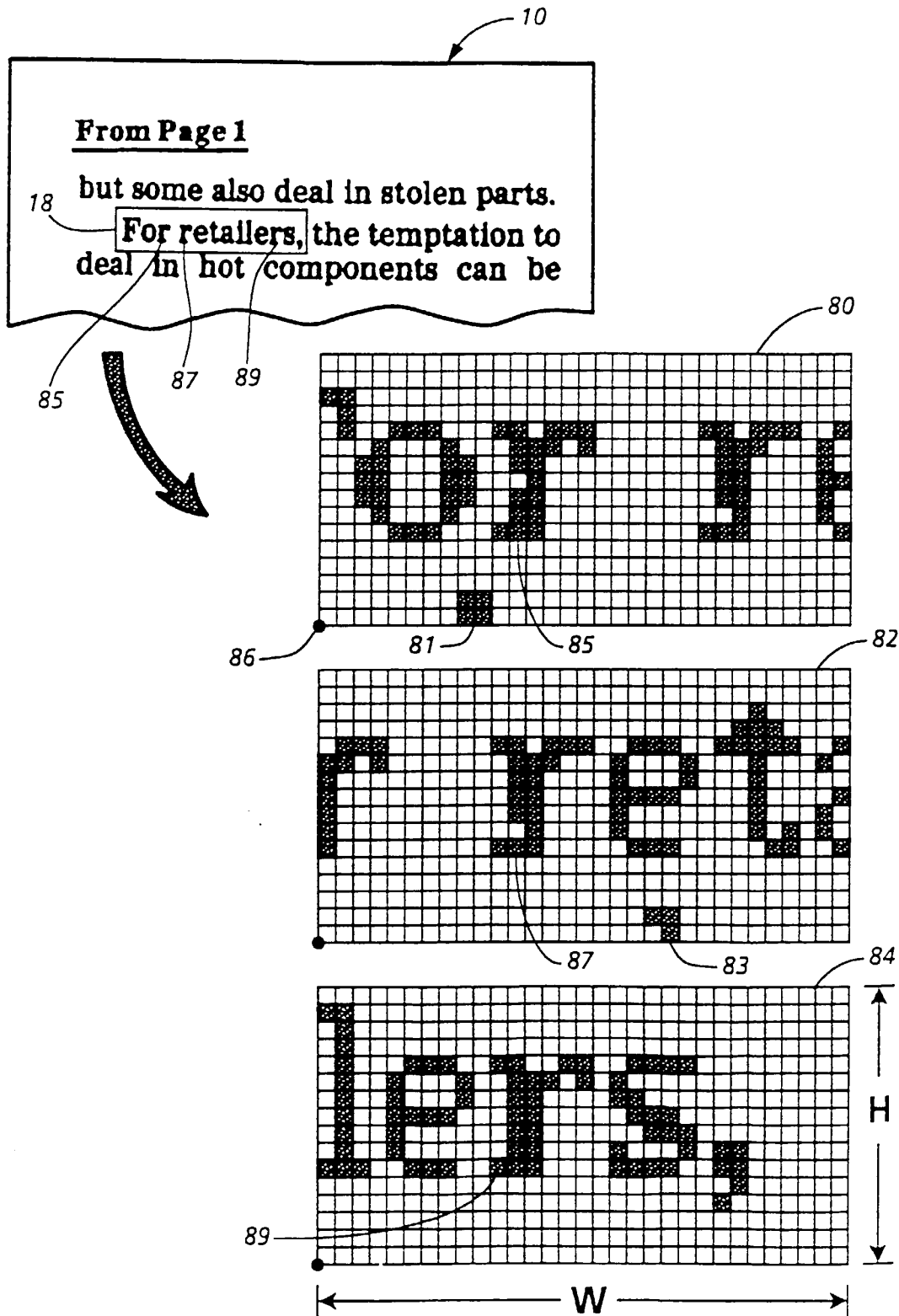


Fig.23

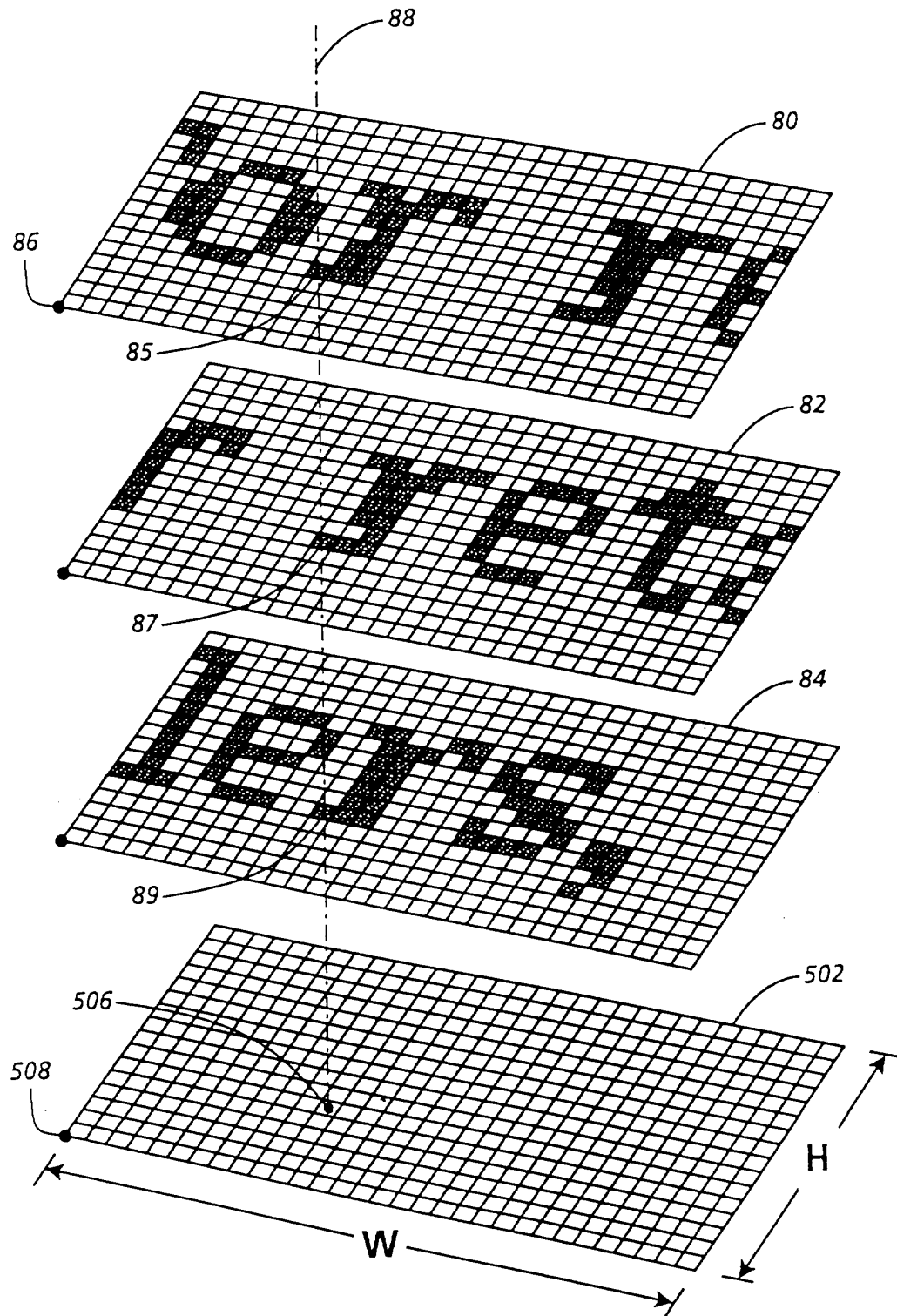


Fig.24

on a rs are cause said, bbage d say, ip and stand wantmpany
 year reveal huctant are uufactu manu part, search mark analys
 an anal, an abbage Paul ch as small erializmpanie nal screase
 treat vrs are uufactu Manu rnel S payir are obably chann that
 t say said: analyst ana Dataqu Datacbbage, mate tet affe mark
 gray rexactio say cular ca partioit a pat's a loct at- lara C: Clara
 anta Cl Sante gang head (Soarer Dave reallyu can r that y: gang
 Asian as a mg as e's anylara. Clara anta l Sante and San
 tigator acca gangAsian) have pear ts app years nt armollars
 ousands andpitas, se and, San Jlara, S Clara, anta Cl Sante 10 arm
 rganis man led a mce arre, San J daylbroad of raids attack by atta
 years past plagu have Bay l Valleypears at app that a natio
 is a have d ages andth abornia, r Califp and brancsuranc
 lstate it at rglarieff a sers are anduufactu manu trackies and
 erial- at seri that smpanyx-basece a that tHe alsoen and may
 plastifera- tags npany anage manant, ad- ultant, Parkerializa
 mpanie al c., said Dataqu Datac Trantet Jami, said mark gray
 in andny are many Stateslegallys cann Mark geant lara po Clara
 anta Sante, accof a sff a shi gangorganis parts made actually
 is actur seas uufactu manu tabl madey are laws paterly lax

Fig.25

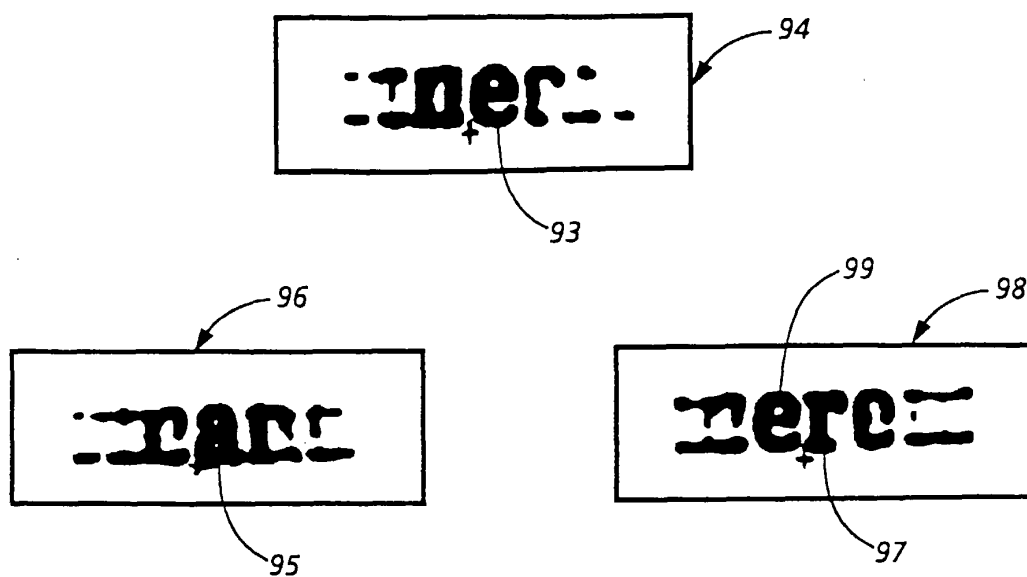
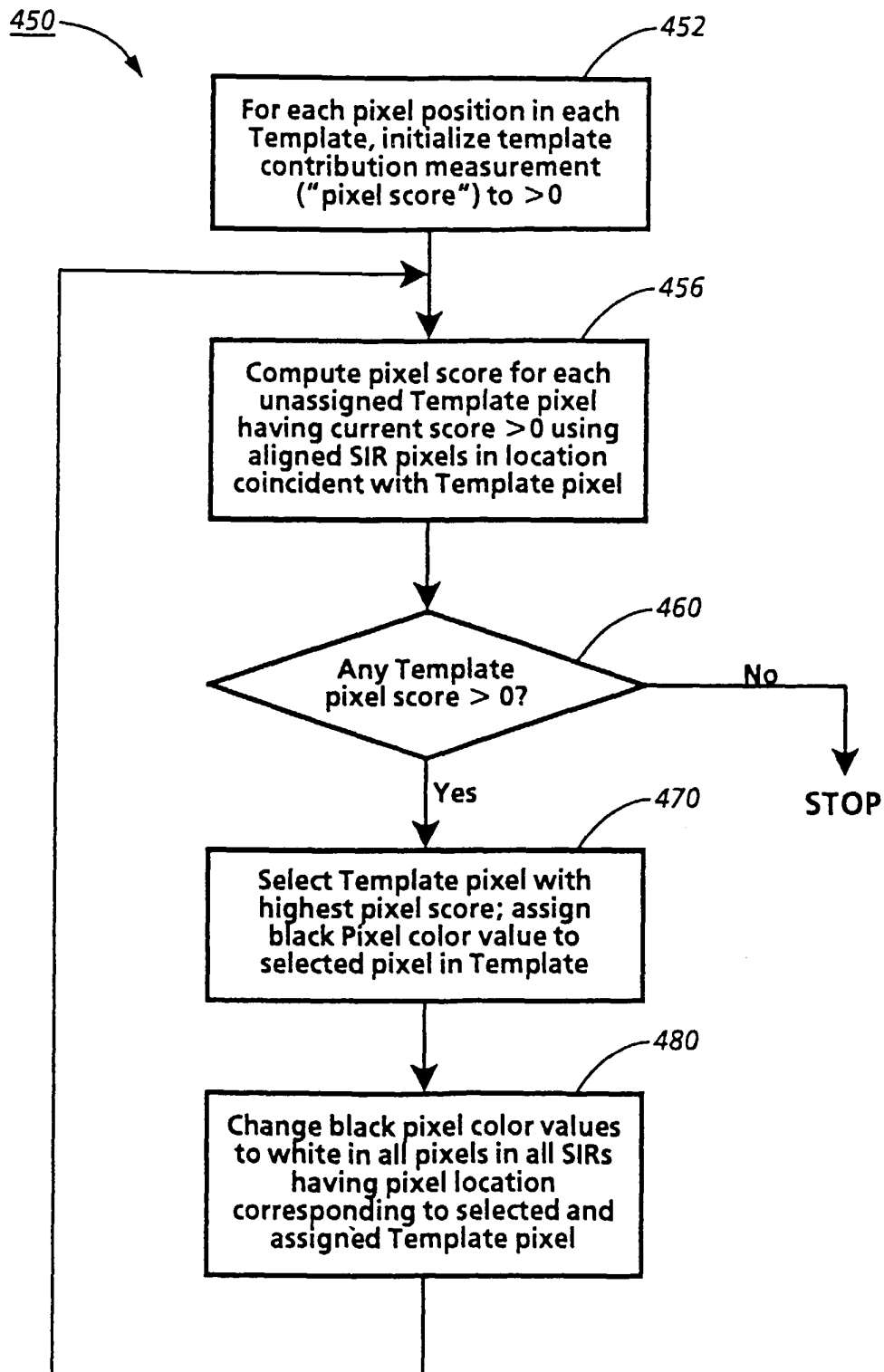


Fig.26

510

520	514				516	518						
I	a	b	c	d	e	f	g	h	i	j	k	
l	m	n	o	p	q	r	s	t	u	v	w	
x	y	z	A	B	C	D	E	F	G	H	I	
J	K	L	M	N	O	P	Q	R	S	T	U	
V	W		Y		1	2	3	4	5	6	7	
8	9	0	.									
,	.				:							

Fig.28

**Fig.27**

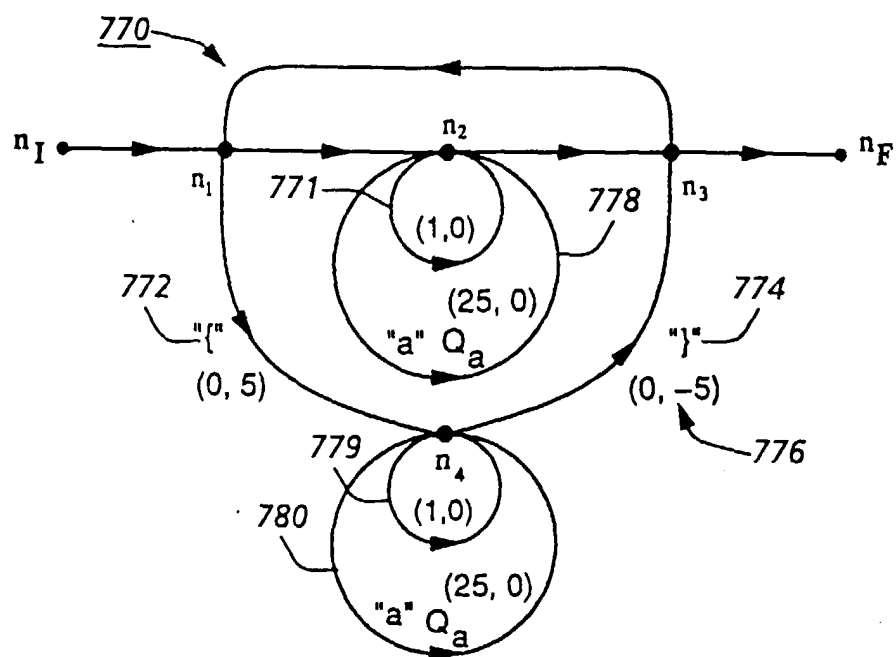


Fig. 29

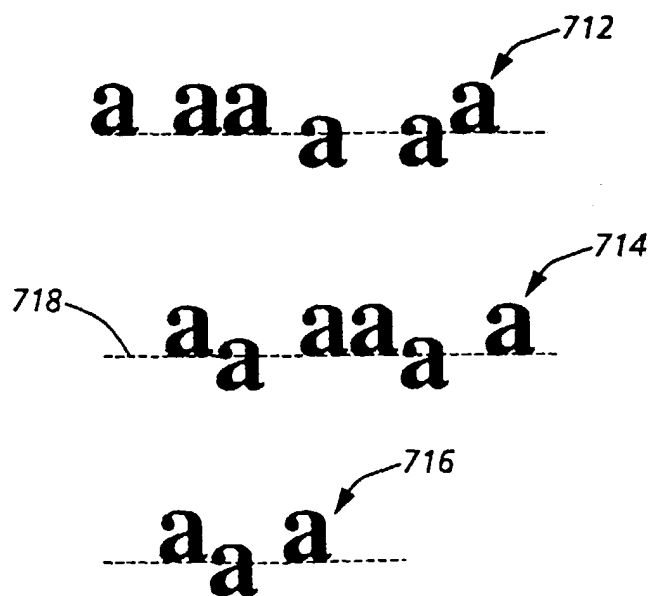


Fig. 30